WebSphere Message Broker

Troubleshooting

Version 6 Release 0

WebSphere Message Broker

Troubleshooting

Version 6 Release 0

Note

Before using this information and the product it supports, read the information in the Notices appendix.

First Edition (September 2005)

This edition applies to IBM® WebSphere® Message Broker Version 6.0 and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

About this topic collection

This PDF has been created from the WebSphere Message Broker Version 6.0 GA (September 2005) information center topics. Always refer to the WebSphere Message Broker online information center to access the most current information. The information center is periodically updated on the document update site and this PDF and others that you can download from that Web site might not contain the most current information.

The topic content included in the PDF does not include the ″Related Links″ sections provided in the online topics. Links within the topic content itself are included, but are active only if they link to another topic in the same PDF collection. Links to topics outside this topic collection are also shown, but these attempt to link to a PDF that is called after the topic identifier (for example, ac12340_.pdf) and therefore fail. Use the online information to navigate freely between topics.

Feedback: do not provide feedback on this PDF. Refer to the online information to ensure that you have access to the most current information, and use the Feedback link that appears at the end of each topic to report any errors or suggestions for improvement. Using the Feedback link provides precise information about the location of your comment.

The content of these topics is created for viewing online; you might find that the formatting and presentation of some figures, tables, examples, and so on are not optimized for the printed page. Text highlighting might also have a different appearance.

Part 1. Troubleshooting

Troubleshooting and support

This section contains information about the various techniques that you can use to help you to diagnose any problems that you are having with WebSphere Message Broker. This section covers the following topic areas:

- "Making initial checks"
- "Using logs" on page 104
- "Using trace" on page 109
- "Using dumps" on page 120
- "Dealing with problems" on page 11
- "Recovering after failure" on page 122
- "Contacting your IBM Support Center" on page 129

You can also read the general troubleshooting guidance in the following topics:

- Troubleshooting overview
- Searching knowledge bases
- Getting fixes
- IBM Software Support

For information specific to debugging message flow applications, see Testing and debugging message flow applications.

Making initial checks

Before you start problem determination in detail, it is worth considering whether there is an obvious cause of the problem, or an area of investigation 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 fundamental questions that you need 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 could be useful later if you have to carry out a systematic problem determination exercise.

- "Has WebSphere Message Broker never run successfully before?" on page 4
- "Did you log off Windows while WebSphere Message Broker components were active?" on page 4
- "Are the Linux andUNIX environment variables set correctly?" on page 4
- "Are there any error messages or return codes that explain the problem?" on page 5
- "Can you reproduce the problem?" on page 6
- "Has the message flow run successfully before?" on page 6
- "Have you made any changes since the last successful run?" on page 7
- "Is there a problem with descriptive text for a command?" on page 8
- "Is there a problem with a database?" on page 8
- "Is there a problem with the network?" on page 8
- "Does the problem affect all users?" on page 9

- "Have you recently changed a password?" on page 9
- "Have you applied any service updates?" on page 9
- "Do you have a component that is running slowly?" on page 9
- "Additional checks for z/OS users" on page 10

Has WebSphere Message Broker never run successfully before?

If so, determine whether your problem is caused by incorrect installation and setup, or whether it was caused by something else.

If it has not run successfully before, complete the following steps.

1. **Check your setup**

   It is likely that you have not set up WebSphere Message Broker correctly.

   On UNIX operating systems, check that the sample profile has been included in your user environment. You can use the sample profile itself, or use the sample as a model, and include the relevant pieces in your user's `.profile` file.
2. **Verify the installation**

   Check the Installation Guide for information about how you can verify a basic configuration on your system. On Linux (X86 platform) and Windows, you can use the Default Configuration wizard and the Samples Preparation wizard to help you with basic configuration and verification.

   For more detailed configuration information, refer to Configuring the broker domain.

Did you log off Windows while WebSphere Message Broker components were active?

On Windows, logging off when WebSphere Message Broker components (a broker, the Configuration Manager, or the User Name Server) are active can cause a problem. You might see messages, including 2070, 2642, 1102, and 1103 in the Windows Event log if this is the case.

When you log off, any queue managers that support these WebSphere Message Broker components are stopped unless they are defined to run as Windows services. The components all run as Windows services and remain active, but they find that the queue manager and queue manager objects associated with them are no longer available.

1. If you log off, restart the queue manager that supports the Configuration Manager or the User Name Server, and that component reestablishes the connection to the queue manager.
2. To avoid this problem, set the queue managers used by the broker, Configuration Manager and User Name Server to run as Windows services, so that logging off Windows does not cause this problem.

Are the Linux andUNIX environment variables set correctly?

A sample profile, which contains the correct environment variable settings for Linux and UNIX systems, is supplied with WebSphere Message Broker in the *install_dir*/sample/profiles directory.

The environment variables that you need to set are listed below; use the sample profile to ensure that your environment variables are set correctly. Also, to ensure that you get the latest environment variables, read the product readme.html file.

- CLASSPATH
- LANG
- LC_ALL
- LD_LIBRARY_PATH
- LC_MESSAGES
- MQSI_REGISTRY
- ODBC_INI
- NLSPATH
- PATH

For information about using the profile, refer to the instructions about creating a broker on your operating system:

- AIX
- HP-UX
- Linux
- Solaris

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**

  WebSphere Message Broker error messages have the prefix BIP. If you receive any messages with this prefix (for example, in the UNIX, Linux, or z/OS syslog), see Diagnostic messages for an explanation.

  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.

  WebSphere Message Broker messages that have a mixture of identifiers such as BIPV600, 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 you resolve the problem.

  Messages that are associated with the startup of WebSphere Message Broker, 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 WebSphere Message Broker broker domain might require some additional configuration of the environment beyond what is recommended in Installing complementary products. The need for such configuration changes are typically indicated by warning or error messages logged by the various components, including WebSphere MQ, the databases, and the operating system. These messages are normally accompanied by a suggested User Response.

Can you reproduce the problem?

If you can reproduce the problem, consider the conditions under which you can do so:

- **Is the problem caused by a particular message flow?** If so, use the debugger facility of the workbench (see Testing and debugging message flow applications) and user tracing to home in on the problem.
- **Is the problem caused by a command?** On distributed platforms, you issue commands at the system command line.

  On z/OS, you can issue commands from the console, the syslog, or by submitting a batch job. You enter customization commands from an OMVS session. Console commands that you enter from the console or syslog might be converted to uppercase, depending on the system configuration. This can cause some commands, such as **mqsichangetrace**, to fail, especially if these commands contain parameters that must be lowercase. An error message indicating that the execution group is not available might be caused by the execution group name being in the wrong case. The same can happen on message flows.
- **Does a problem command work if it is entered by another user ID?**

  If the command works when it is entered by another user ID, check the environment of each user. Paths, especially shared library paths, might be different. Also, verify that all users have included the sample profile provided with WebSphere Message Broker; see "Are the Linux andUNIX environment variables set correctly?" on page 4 for more information about this sample. On distributed platforms, the environment includes:
  - Group memberships
  - Environment variables
  - Shell type (on Linux and UNIX)

  On Windows, the environment for the broker comes from the system settings, not from a particular user's variables. However, the user's variables affect non-broker commands.

  On Linux and UNIX systems, only the service ID specified when the broker was created can start a broker.

  On Windows, any authorized user can start a broker.

Has the message flow run successfully before?

Sometimes a problem will appear in a message flow that has previously run successfully. To identify the cause of the problem, ask yourself the following questions:

- **Have you made any changes to the message flow since it last ran successfully?**

  If so, it is likely that the error lies somewhere in the new or modified part of the flow. Examine the changes and see if you can find an obvious reason for the problem.
- **Have you fully used all the functions of the message flow before?**

  Did the problem occur when you used part of the message flow that had never been invoked before? If so, it is likely that the error lies in that part. Try to find out what the message flow was doing when it failed to do this; use user tracing, trace nodes, and the workbench's debugger function (see Debugging a message flow).

If you have run a message flow successfully on many previous occasions, check the current queue status and the files that were being processed when the error occurred. It is possible that they contain some unusual data value that invokes a rarely-used path in the message flow.

- **Does the message flow check all return codes?**

  Could it be that your system has been changed, perhaps in a minor way, but your message flow does not check the return codes it receives as a result of the change? For example:

  - Does your message flow assume that the queues it accesses can be shared? If a queue has been redefined as exclusive, can your message flow deal with return codes indicating that it can no longer access that queue?
  - Have any security profiles been changed? A message flow could fail because of a security violation.
- **Does the message flow expect particular message formats?**

  If a message with an unexpected message format has been put onto a queue (for example, a message from a queue manager on a different platform) it might require data conversion or a different form of processing. Also, check whether you have changed any of the message formats used.
- **Does the message flow run on other WebSphere Message Broker systems?**

  Is there something different about the way that your system is set up that is causing the problem? For example, have the queues been defined with the same maximum message length or priority? Are there differences in the databases used, or their setup?

  If a message flow has not run successfully before, examine your message flow carefully to see if you can find any of the following errors:
- **Are you using any user-defined extensions?**

  There might be translation or compilation problems with loadable implementation library (LIL) files. Before you look at the code, examine the output from the translator, the compiler or assembler, and the linkage editor, to see if any errors have been reported. Fix any errors to make the user-defined extension work.

  If the documentation shows that each of these steps was completed without error, consider the coding logic of the message flow, message set, or user-defined extension. Do the symptoms of the problem indicate the function that is failing and, therefore, the piece of code in error? See User-defined extensions for more information.
- **Do you see the errors from WebSphere Message Broker or external resources, such as databases?**

  Your message flow might be losing errors because of incorrect use of the failure terminals on primitive nodes. If you use the failure terminals, make sure that you handle errors adequately. See Handling errors in message flows for more information about failure terminals.

Have you made any changes since the last successful run?

When you are considering changes that might recently have been made, think not only about WebSphere Message Broker, but also about the other programs it interfaces with, the hardware, device drivers, and any new applications.

- **Have you changed your initialization procedure?**

  On z/OS, have you changed any data sets or library definitions? Has the operating system been initialized with different parameters? Check for error messages generated during initialization.

- **Has the profile of the user running the commands on Linux or UNIX operating systems been changed?**

  If so, this might mean that the user no longer has access to the required objects and commands.
- **Has any of the software on your system been upgraded to a later release?**

  Check that the upgrade completed successfully, and whether the new software is compatible with WebSphere Message Broker (check the product readme.html file).
- **Do your message flows deal with the errors and return codes that they might get as a result of any changes that you have made?**

  Check that your message flow can handle all possible error situations.

Is there a problem with descriptive text for a command?

On Linux and UNIX systems, be careful when including special characters (back slash (\) and double quote (″) characters) in descriptive text for some commands. If you use either of these characters in descriptive text, precede them with a \, that is, enter \\ or \″ if you want \ or ″ in your text.

Is there a problem with a database?

If you have database problems, make the following checks:

- Check that the database is started.
- Check that your ODBC.ini file on Linux or UNIX operating systems (if you are using the default ODBC.ini file, this is located at <*install_dir*>/odbc/.odbc.ini).

  For a broker database, the basic configuration is correct if the broker can be created. If the broker can be created, but not started, use ODBC tracing to investigate the cause.
- Check the number of database connections: on DB2 for AIX, using local mode connections, the maximum is 10.
- For messages indicating that imbdfdb2v6.lil (the 32-bit XA support LIL) failed to load, ensure that a supported database is installed.

Is there a problem with the network?

WebSphere Message Broker uses WebSphere MQ for inter-component communication. If components are on separate queue managers, they are connected by message channels, although communications between the workbench and the Configuration Manager use MQI channels.

There can be communication problems between any of these:

- Brokers
- The User Name Server
- The Configuration Manager
- The workbench

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.

Does the problem affect all users?

On distributed operating systems different users have different environments, and this might explain some problems. Check whether there is a different user ID in an incoming message, or a different user ID running a command (or acting as the broker's service ID).

Have you recently changed a password?

If you have changed the password at the operating system level for the user ID that is used either as the ServiceUserId or the DataSourceUserId for any of the WebSphere Message Broker components, this might cause a problem. If so, perform the following steps:

- Change the properties of the affected component to reflect the password change. Use the **mqsichangebroker**, **mqsichangeconfigmgr**, and **mqsichangeusernameserver** commands to change the properties of the WebSphere Message Broker components.
- Make sure that your passwords do not contain reserved keywords. For example, IBM, which is a reserved keyword in DB2.

Have you applied any service updates?

If an APAR or PTF has been applied to WebSphere MQ for z/OS, or a fix pack or interim fix has been applied to a distributed operating system, check that no error message was produced. If the installation was successful, check with the IBM Support Center for any known error with the service update.

- If a service update has been applied to any other product, consider the effect it might have on WebSphere Message Broker.
- Ensure that you have followed any instructions in the service update that affect your system. (For example, you might need to redefine a resource.)
- If you are not sure whether a service update has been applied to your system, check the memo.ptffile. This file contains the service level and details of the maintenance supplied.
- To view the memo.ptf file on Windows, click **Start** → **Programs** → **WebSphere Business Integration Message Broker** → **Service level**.

Do you have a component that is running slowly?

If a particular component, or the system in general, is running slowly, complete the following steps:

1. Check whether you have inadvertently left tracing on. This could be WebSphere Message Broker user tracing or service tracing, ODBC tracing, WebSphere MQ tracing, or native database tracing. If you have, turn the tracing off.
2. Clear out any old abend files from your errors directory. If you do not clear it of unwanted files you might find that your system performance degrades due to significant space being used up.

3. On Windows, use the workpath **-w** flag of the **mqsicreatebroker** command to create the errors directory in a hard drive partition that does not contain WebSphere Message Broker or Windows.
4. Increase your system memory.

Additional checks for z/OS users

On WebSphere Message Broker for z/OS you might find that:

- The console has unexpected messages; look for an explanation in Diagnostic messages or other relevant messages and codes documentation.
- RRS is not started; issue the command D A,RRS.
- The queue manager is not available; issue the WebSphere MQ command DISPLAY THREAD(*).
- DB2 is not working; issue the DB2 command DISPLAY THREAD.
- The User Name Server is not working.

The cause of your problem on z/OS could be in any of the following areas. Check each of these, starting with whichever seems the most likely, based on the nature of your problem.

- The queue manager address space.
- A queue manager in your queue-sharing group.
- The channel initiator address space.
- Batch or TSO address space.
- The z/OS system (including ARM, RRS, or the Coupling Facility).
- The network (including APPC or TCP/IP).
- Another system, for example a queue manager on another operating system or a WebSphere MQ client.
- The external security manager product, for example RACF or ACF2.
- DB2

Understanding interactions between the runtime components

There are three runtime components: the broker, the Configuration Manager and the execution group. These components communicate with each other by exchanging command requests inside WebSphere MQ messages to perform actions, such as deploying a message flow. After completion of the command request, responses are sent back to the originating component, indicating whether the request was successful.

On WebSphere Message Broker for z/OS, you will see extra information messages issued by z/OS runtime components that allow you and IBM Service personnel to determine the interactions between the various runtime components, including the Configuration Manager, broker and execution group.

When the broker receives command requests from the Configuration Manager, the broker issues a message that identifies the command request. When this request completes, a message is issued that indicates whether the command is successful. Each command request that the broker receives results in at least one matching command request being sent to an execution group, and a corresponding message being issued. Every response from an execution group that results from these command requests will result in a message being issued. These messages can be turned on and off and are to be used by the IBM Service team.

When an execution group receives command requests from the broker, the execution group issues information messages that identify the command request. For actions contained within a command request, an information message is issued that identifies the action to be performed and the resource upon which the action is to take effect. When the request completes, the execution group issues a message that indicates that the message has been processed.

When a Configuration Manager receives a command request and when this request completes, information messages are issued.

Loading IBM and user-defined nodes and parsers

When an execution group starts, it loads all available IBM and user-defined nodes and parsers. For each library that is loaded, two messages are issued. One is issued before the library is loaded and one is issued after the library has been loaded.

Dealing with problems

Before you start:

Make initial checks and check any relevant logs, trace and dumps to see if you can rectify the problem.

This section describes some of the typical problems that can occur, and what actions you can take to resolve them. It contains the following topics:

- “Resolving problems when creating resources”
- “Resolving problems when starting resources” on page 16
- “Resolving problems when importing or migrating resources” on page 27
- “Resolving problems when stopping resources” on page 32
- “Resolving problems when deleting resources” on page 33
- “Resolving problems when developing message flows” on page 36
- “Resolving problems when deploying message flows or message sets” on page 48
- “Resolving problems when debugging message flows” on page 59
- “Resolving problems when developing message models” on page 64
- “Resolving problems when using messages” on page 70
- “Resolving problems when using the workbench” on page 77
- “Resolving problems when using databases” on page 82
- “Resolving problems when using publish/subscribe” on page 89
- “Resolving problems with performance” on page 95
- “Resolving problems when developing Configuration Manager Proxy applications” on page 98
- “Resolving problems with user-defined extensions” on page 100

Resolving problems when creating resources

This topic contains advice for resolving problems that can occur when you are creating resources.

Problems when creating a broker:

Problems when creating other resources:

Messages BIP0874E and BIP1776S are issued when creating a broker

- **Scenario:** Error messages BIP0874E and BIP1776S are displayed when you try to create a broker.
- **Explanation:** This problem occurs if the IBM Agent Controller was not installed correctly before the Configuration Manager was created.
- **Solution:** If this is the case:
  1. Delete the Configuration Manager using the **-n** and **-m** flags with the **mqsideleteconfigmgr**command to remove all data.
  2. Delete the Configuration Manager database.
  3. Install the IBM Agent Controller.
  4. Re-create the Configuration Manager database.
  5. Re-create the Configuration Manager.

     You can now create a broker.

Message BIP1511Eis issued when creating a broker

- **Scenario:** Error message BIP1511E is displayed when you try to create a new broker in the Topology editor.

  ```
  BIP1511E: Queue manager 'QM1' cannot be associated with broker
  'BR1'; it is already associated with broker 'BR1'
  ```
- **Explanation:** A queue manager cannot be associated with a broker because it is already associated with a broker. The problem occurs because you performed an incorrect sequence of actions when you deleted the broker, and you are now trying to re-create a broker of the same name.
- **Solution:** Delete the broker before attempting to re-create it:
  1. Stop the broker using the **mqsistop** command.
  2. Delete the broker from the topology.

3. Check that the delete was successful, and that the broker has disappeared from the workbench.
4. If you are using publish/subscribe and the network of connected brokers has been modified as a result of the deletion, issue a delta deploy of the topology.
5. Delete the broker using the **mqsideletebroker**command.

Do not try to re-create the broker until the deleted topology has been successfully deployed. If the broker is no longer shown in the workbench, you can then re-create the broker, as follows:

1. Create the broker using the **mqsicreatebroker**command.
2. Create the broker in the topology.
3. Deploy the topology.

Message BIP0874E is issued while saving the Topology editor after creating or deleting a broker

- **Scenario:** Message BIP0874E is displayed while you are saving the Topology editor after creating or deleting a broker, and the broker is not created or deleted.
- **Explanation:** The problem occurs because you are no longer authorized to modify the topology.
- **Solution:** Ask your WebSphere Message Broker administrator to give the user ID full access to the broker topology.

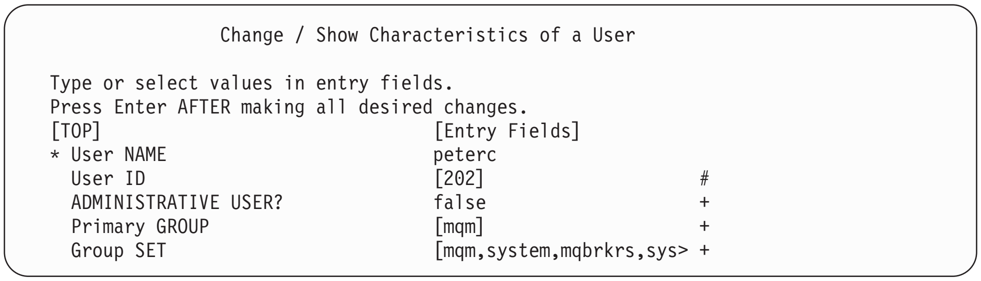
  If you close the editor and save the topology changes, you do not receive any error messages.

You cannot create files when creating a broker on AIX

- **Scenario:** You run the **mqsicreatebroker** command on WebSphere Message Broker for AIX, and the following message is displayed:

  ```
  BIP8135E Unable to create files. Operating System return code 1
  ```

- **Explanation:** The user ID that you create for WebSphere Message Broker testing must have a primary group of **mqm**, and have **mqbrkrs** as one of the group set. The following is an example of a SMIT panel on AIX showing the change and show characteristics of a user:



```
                 Change / Show Characteristics of a User

 Type or select values in entry fields.
 Press Enter AFTER making all desired changes.
 [TOP]                              [Entry Fields]
 * User NAME                         peterc
   User ID                           [202]                #
   ADMINISTRATIVE USER?              false                +
   Primary GROUP                     [mqm]                +
   Group SET                         [mqm,system,mqbrkrs,sys> +
```

The Configuration Manager does not remove references to a deleted broker’s queue manager

- **Scenario:** The broker definition is incorrect.
- **Explanation:** This can occur if you delete a broker component without removing it from the Message Brokers Toolkit first. If your broker definition is incorrect, you cannot instruct the Configuration Manager to add a new broker with the same queue manager to the broker domain because the Configuration Manager reports that the queue manager is already associated with the broker that you deleted.

If your broker definition is incorrect, you cannot redeploy. If you delete a broker component then try to deploy to it, any subsequent deploy to that broker fails because the Configuration Manager reports that it is already deploying to the broker that you have deleted.

- **Solution:** Delete the queue manager for the deleted broker then deploy. When you deploy, the Configuration Manager recognizes that the queue manager does not exist and cleans up the broker's definition. You can now associate the queue manager with another broker or redeploy the broker.

The JCL BIPGEN fails when you create a component on z/OS

- **Scenario:** BIPGEN fails when you are copying the component profile (BIPBPROF, BIPCPROF, or BIPUPROF) from the PDSE to the HFS.
- **Explanation:** There might not be enough space in the HFS, the component profile might not exist, or you might not have the appropriate authority.
- **Solution:** Make the following checks:
  - Check that the HFS has sufficient space. You can check how much space is used and how much is free in an HFS using the OMVS command `df -P /pathname`. 100 MB is 3 276 800 512 byte sectors.
  - Check that the profile file exists in the PDSE.
  - Check that you have the appropriate authority to write to the HFS.

The mqsicreatebroker command hangs if the DB2 profile has not been run

- **Scenario:** The **mqsicreatebroker** command hangs if the DB2 profile has not been run. If you stop the command, this could result in a partial broker setup. This partially-created broker might be shown by **mqsilist** and the registry but none of the broker tables will have been created.
- **Explanation:** This error can occur if the environment has not been set up correctly.
- **Solution:** Remove any partial setup and ensure that the environment is set up correctly by following the instructions in Configuring the broker domain. Run the DB2 profile before re-issuing the **mqsicreatebroker** command.

Error message BIP8075E is issued when creating a Configuration Manager

- **Scenario:** On a Windows system, the **mqsicreateconfigmgr**command fails with message BIP8075E and a Java exception unsatisfiedLinkException.
- **Explanation:** The Configuration Manager cannot find the JAR files that it needs to connect to the configuration database. If you have installed additional software since installing the broker, or have made manual updates to your CLASSPATH, these might cause the **mqsicreateconfigmgr**command to fail.
- **Solution:** Ensure that the DB2 JAR files, especially db2java.zip, are in your CLASSPATH.

Error message BIP2624 is issued when creating an execution group

- **Scenario:** When you create an execution group, you get several BIP2624 messages that indicate MQRC=2012 (MQRC_ENVIRONMENT_ERROR), and no WebSphere MQ messages are processed.
- **Explanation:** You have created the broker to run as a WebSphere MQ trusted application (that is, the broker runs in the same process as the WebSphere MQ queue manager), but the user ID that you specified does not have the required authority.

- **Solution:** If you request the trusted application option on the **mqsicreatebroker** command by specifying the flag **-t**:

  Windows
  Specify a service user ID that is a member of WebSphere MQ group **mqm** on the **-i** flag on the **mqsicreatebroker** command.

  AIX The **-t** flag is ignored. The broker does not run as a trusted application on AIX.

  Other UNIX platforms
  Specify the user ID **mqm** on the **-i** flag on the **mqsicreatebroker** command.

You do not know what authorities are set as part of the mqsicreateaclentry command

- **Scenario:** You do not know what authorities are set on the queue manager as part of the **mqsicreateaclentry**command.
- **Solution:** Use the WebSphere MQ **dspmqaut** command to check which authorities have been set on a queue manager by the **mqsicreateaclentry** command. The following authorities should be set:
  - inq
  - set
  - connect
  - altusr
  - chg
  - dsp
  - setall

Error message BIP0889E is issued when creating a domain connection

- **Scenario:** When you create a domain connection, it fails with the following error:

  ```
  BIP0889E An exception occurred while communicating with
  Configuration Manager on <Queue Manager>.
  Reason:
  WebSphere studio is experiencing problems communicating with the
  configuration manager on <Queue Manager>

  Ensure that the configuration manager is available
  If so and exception still occurs, retry the operation after
  increasing configuration manager proxy retry attempts value in
  preferences
  ```

  Details similar to the following are given:

  ```
  PubSubTopology information was not received from the
  Configuration Manager. Either the Configuration Manager is not
  available or the user '<username>' does not have authority to
  view the object
  (UUID='PubSubTopology', required attribute='subcomponent.last')
  ```

- **Explanation:** The WebScale Distribution Hub, which is installed as part of a full WebSphere Message Broker installation, has not installed correctly.
- **Solution:** Check that the distribution hub installation directory C:\Program Files\IBM\Disthub\v2 contains folders called classes and lib. If the directory contains only an _uninstall folder, you need to uninstall disthub, and install it again, as described below:
  1. At a command prompt, change the directory to C:\Program Files\IBM\Disthub\v2_uninst.

2. Run java -jar uninstall.jar.
3. To complete the uninstallation, select options **1**, **1**, and **3**.
4. The product installation package contains a folder called disthub. At a command prompt, change to this folder.
5. Run java -jar setupwin.jar.
6. Select options **1**, **1**, and **3** to complete the installation of the distribution hub.

Resolving problems when starting resources

This topic contains advice for dealing with some common problems that can arise when starting resources:

- “Resources terminate during startup” on page 17
- “Resources hang at startup on Windows” on page 17
- “Error message BIP8048 is issued when you start a component.” on page 17
- “Error message BIP2228E is issued when you try to start a second broker on Linux or UNIX” on page 18
- “Broker and execution group are not running after using the Default Configuration wizard” on page 18
- “You get a ″Not Found″ error when you click a link to the Samples Gallery, or to a specific sample” on page 18
- “Database error message BIP2322E is issued when you start or stop a broker on Linux or UNIX” on page 19
- “Error message BIP832E is issued on startup” on page 19
- “The broker or Configuration Manager fails to start because there is not enough space in the Java TMPDIR directory” on page 19
- “A broker fails to start on z/OS” on page 19
- “You get abend code 0C1 when trying to start the data flow engine on z/OS” on page 20
- “Your execution groups restart repeatedly” on page 20
- “You cannot tell whether startup is complete on z/OS” on page 20
- “You cannot start the User Name Server” on page 21
- “You have changed your logon password and cannot start your Configuration Manager or broker” on page 21
- “MQIsdp client connection is refused by broker” on page 22
- “You get abend code 047 and a diagnostic message” on page 22
- “You get abend code 047 in bin/bipimain” on page 23
- “Error message BIP2048E is issued on z/OS when your broker fails to start” on page 23
- “Error messages BIP2604 or BIP2624 are issued when you start a broker or a new message flow” on page 24
- “Error message BIP2604 with return code MQRC_CONNTAG_IN_USE is issued during the start of a message flow on z/OS” on page 24
- “A device allocation error is issued” on page 25
- “Diagnostic message ICH408I is issued on z/OS when your broker fails to start” on page 25
- “When you start the broker through the Dataflowengine, it cycles continually” on page 26
- “The create command fails and error message BIP8022E is issued” on page 26

- “Broker terminates with an abend on startup on AIX” on page 26
- “Error code BIP0889E is issued when you connect to the Configuration Manager” on page 27
- “Connection to the Configuration Manager is slow” on page 27
- “You experience problems connecting to the Configuration Manager queue manager” on page 27

Resources terminate during startup

- **Scenario:** The following error message is displayed when you start a broker, the Configuration Manager, or the User Name Server on Windows:

  ```
  ServiceName - DLL initialization failure Initialization of the
  dynamic link library c:\windows\system32\user32.dll failed.
  The process is terminating abnormally.
  ```
- **Explanation:** This error message is issued by Windows when it fails to start a service because it has insufficient storage.
- **Solution:** This is an operating system problem. Information on how to recover from this problem is available in the Microsoft Developer Network (MSDN). You can access MSDN on the Web at:

  ```
  http://msdn.microsoft.com
  ```

Resources hang at startup on Windows

- **Scenario:** You try to start the Configuration Manager or a broker on Windows, but nothing happens in the Event log to show that it has started connecting.
- **Explanation:** This problem is caused by processes having only one thread. To see if that is the case, check the Windows Task Manager. If either of the processes bipconfigmgr.exe or dataflowengine.exe has started, check the number of threads owned by the process. If the process has only one thread, you probably have this problem.
- **Solution:**
  1. Shut down the broker using the **mqsistop** command and end the process from within the Task Manager.
  2. Click **Settings** → **Control Panel** → **Services** and make a note of the current settings.

     Attention:

     You lose the **This Account**,**Password**, and **Confirm Password** entries when you make these changes, so make sure that you know those details before you click **OK**.
  3. Select **System Account** as the **Log On As** option, and tick the box **Allow Service to Interact with Desktop.** This allows you to see any hidden dialog messages.
  4. Restart the resource that is failing, and report any subsequent error messages and dialog box messages to your IBM Service Representative.
  5. When your IBM Service Representative has resolved this problem for you, make sure that you restore the **This Account**, **Password**, and **Confirm Password** entries to those that you used when you created the broker.

Error message BIP8048 is issued when you start a component.

- **Scenario:** Error message BIP8048 is issued when you start a component.
- **Explanation:** This message indicates that WebSphere MQ is not responding as expected when it attempts to start the queue manager. This could be because the

strmqm executable file is not present on UNIX or Linux, or the **amqmdain** executable file is not present on Windows, or permissions are incorrect.

- **Solution:** Check that your WebSphere MQ installation is fully functional:
  - On Windows, start the ″IBM MQSeries″ service.
  - On UNIX or Linux, run the **strmqm** command to start the queue manager associated with this component.

  If the check fails, your WebSphere MQ installation is incomplete. This is typically because you have previously installed WebSphere Application Server, which installs an embedded WebSphere MQ component that does not support WebSphere Message Broker.

  You must uninstall WebSphere Application Server, then install the full WebSphere MQ product that is provided with WebSphere Message Broker.

Error message BIP2228E is issued when you try to start a second broker on Linux or UNIX

- **Scenario:** Error message BIP2228E , mentioning `semctl` in the syslog, is displayed when you try to start a second broker on Linux or UNIX.
- **Explanation:** These error messages usually indicate a permissions problem with a semaphore used by WebSphere Message Broker. A semaphore is created when the first broker starts after a reboot (or after an initial installation), and only members of the creator of the semaphore's primary group can access this semaphore. This is a consequence of the UNIX System V IPC primitives used by WebSphere Message Broker.

  The BIP2228E message is logged by any broker that is started by a user who is not a member of the semaphore creator's primary group. The broker tries to access the semaphore, but fails with a permissions-related error. It then terminates with the BIP2228E message.
- **Solution:** Avoid this problem by ensuring that all user IDs used to start WebSphere Message Broker have the same primary group. If this is impractical, ensuring that all user IDs are members of each others' primary groups should be sufficient. Contact your IBM Support Center for further assistance.

Broker and execution group are not running after using the Default Configuration wizard

- **Scenario:** You have used the Default Configuration wizard to create a default configuration, and the alerts in the Broker Administration perspective show that your broker and execution group are not running.
- **Explanation:** At this stage nothing has been deployed to the broker or execution group, so they are displayed as not running. After you have deployed to the broker, the correct state of the broker and execution group are displayed.

You get a ″Not Found″ error when you click a link to the Samples Gallery, or to a specific sample

- **Scenario:** You get a ″Not Found″ error indicating that a URL is invalid when you click a link to the Samples Gallery, or to a specific sample.
- **Explanation:** The Samples Gallery and the sample applications are available only if you are viewing the information center within the Message Brokers Toolkit. If you are viewing a standalone information center you cannot access these resources.
- **Solution:** If you want to access the Samples Gallery, ensure that you are viewing the information center within the Message Brokers Toolkit.

Database error message BIP2322E is issued when you start or stop a broker on Linux or UNIX

- **Scenario:** The following error message is displayed when you start or stop a broker on a Linux or UNIX operating system:

  ```
  BIP2322E: Database Error. DataDirect ODBC driver License file not found
  ```
- **Explanation:** When starting or stopping a broker from a user ID other than the one that created the broker, that is, not the service user ID, a problem might occur where WebSphere Message Broker cannot find the necessary DataDirect license file (*install_dir*/wmqi/IVUK.LIC).
- **Solution:** Insert a link to the DataDirect license file into the /usr/local/libdirectory.

Error message BIP832E is issued on startup

- **Scenario:** The following error message is displayed on startup:

  ```
  BIP832E: A class java.io.FileNotFoundException exception occurred which reported
  the following message:
  [filepath] (The process cannot access the file because it is being used by another process).
  Resolve the reason of error and try again.
  ```
- **Explanation:** An invalid WebSphere MQ Java Client trace output file has been specified on the Enqueue preferences screen.
- **Solution:**
  1. Open the Enqueue preferences screen by clicking **Windows** → **Preferences**, then clicking **Enqueue** on the left.
  2. In the **To file** field, specify a valid output file (one that is not read-only or already in use).

The broker or Configuration Manager fails to start because there is not enough space in the Java TMPDIR directory

- **Scenario:** The broker or Configuration Manager fails to start and you see an error message indicating that there is insufficient space.
- **Explanation:** The broker and Configuration Manager use Java JAR files. When the broker or Configuration Manager starts, the Java runtime unzips the JAR files into a temporary directory, the Java TMPDIR directory. On Linux, UNIX and z/OS systems, the TMPDir is usually /tmp; on Windows systems, it is c:\temp. If this directory is not large enough to hold the JAR files, the broker or Configuration Manager will not start.
- **Solution:** Use one of the following methods to specify the location of this temporary JAR directory:
  - use the environment variable *TMPDIR*
  - set the system property **java.io.tmpdir**.

  Allow at least 50 MB of space in this directory for broker components and 10 MB of space for Configuration Manager components. More space might be needed if you deploy large user-defined nodes or other JARs to the broker component.

A broker fails to start on z/OS

There are a number of potential problems that might arise when trying to start a broker on z/OS and these are described in the following sections.

You get abend code 0C1 when trying to start the data flow engine on z/OS

- **Scenario:** The first two WebSphere Message Broker address spaces start successfully, but the third address space (the data flow engine) fails to start. The result is an 0C1 abend.
- **Explanation:** The dataflow engine address space is generated by the admin agent. If the region size is too small, either because an insufficient region size was specified in the procedure, or the region size was overridden by the z/OS IEFUSI exit, the data flow engine address space might fail to start and fail with an 0C1 abend.
- **Solution:** Check this by using IPCS on the dump (moving the dump from the HFS to a traditional MVS data set if required). The IPCS command is:

  `verbx vsmdata,'noglobal,jobname(`*`vcp0brk2`*`)'`

  where *vcp0brk2* is the name of the failing job. Find `'VSM LOCAL DATA AREA AT ADDRESS '`. The field ELIM gives the available region size, and must be greater than 0C800000. If the field SMFEL is not ffffffff, the IEFUSI exit has changed the allowable region size. This value must also be greater than 0C80000.

  If you have an IEFUSI exit, check that the exit does not limit the broker address spaces. For example, a commonly-used field is OUCBSUBN. This can be STC or OMVS for the broker and indicates how the address space was started.

Your execution groups restart repeatedly

- **Scenario:** Your execution groups restart repeatedly. The syslog might show an error, such as BIP2060E.
- **Explanation:** The problem might be caused by:
  - Incorrect environment for the broker
  - LIL directory permissions
  - Database permissions
  - Invalid user-written LILs
- **Solution:** Check:
  - Environment variables
  - File and directory permissions
  - Group memberships

You cannot tell whether startup is complete on z/OS

- **Scenario:** You cannot tell whether startup has completed on your z/OS system.
- **Solution:** To decide if startup is complete:
  1. Check the messages in the syslog. The following is an example of a syslog entry for a startup of a broker with one execution group.

```
S STU3053
$HASP100 STU3053  ON STCINRDR
IEF695I START STU3053  WITH JOBNAME STU3053  IS ASSIGNED TO USER STU3
 , GROUP STCGROUP
$HASP373 STU3053  STARTED
+(broker53) 0 BIP9141W: The component was started.
+(broker53) 0 BIP2001I: The WebSphere Business Integration Message Broker service has started
 process ID 33554919.
+(broker53.default) 0 BIP2201I: Execution Group started: process '67109
442
 196'; thread '0'; additional information 'broker53', '76eb7f2d-e800-00
 00-0080-974c271866d2', 'default', 'true', 'Q4A3', 'false', 'ARG5D651',
  'ARGO53', '********', 'false', 'f9c27f2d-e800-0000-0080-974c271866d2'
 , '/local/argo/driver/drv3', '/local/argo/tgrp53/broker53'.
```

```
+(broker53.default) 0 BIP9137I: A work manager has been registered by R
443
 RMS registration services, work manager name is BIP.STU30532.006710919
 6.IBM.UA  .
```

2. Display the address spaces. The following example shows the display of a broker with one execution group.

```
D OMVS,U=STU3
BPXO040I 18.49.59 DISPLAY OMVS 446
OMVS     000E ACTIVE          OMVS=(68,05)
USER     JOBNAME  ASID        PID       PPID STATE   START     CT_SECS
STU3     STU30531 0069   33554696   33554919 HR    18.49.15      2.217
  LATCHWAITPID=        0 CMD=bipbroker broker53
STU3     STU30532 03FD   67109196   67109222 HR    18.49.23     19.816
  LATCHWAITPID=        0 CMD=DataFlowEngine broker53 76eb7f2d-e800-00
STU3     STU3053  0036   33554768   83886483 HRI   18.49.08       .653
  LATCHWAITPID=        0 CMD=bipservice Q4A3BRK AUTO
STU3     STU30532 03FD   67109222   33554696 1W    18.49.23     19.816
  LATCHWAITPID=        0 CMD=bipimain DataFlowEngine broker53 76eb7f2
STU3     STU3053  0036   83886483          1 1WI   18.49.08       .653
  LATCHWAITPID=        0 CMD=/local/argo/driver/drv3/bin/bipimain bip
STU3     STU30531 0069   33554919   33554768 1W    18.49.15      2.217
  LATCHWAITPID=        0 CMD=bipimain bipbroker broker53
```

The infrastructure main program bipimain is the first process in every address space. It starts either bipservice, bipbroker, DataFlowEngine, or bipuns as the second process in the same address space. For each execution group, an additional address space is started. In this example, only one execution group is available.

You cannot start the User Name Server

- **Scenario:** You have created the User Name Server specifying the local system name for the Windows security domain, and used the same identifier for the service user ID. When you start the User Name Server, the following message is generated:

  ```
  BIP8026E: Unable to start the component.
  The component could not be started using the service userid that was supplied when the
  component was created. Ensure that the service userid and password are still valid.
  Ensure that the service userid has permission to access all of the products directories,
  specifically the 'bin' and 'log' directories.  Check for system messages (on Windows NT
  this would be the application event log).
  ```

- **Explanation:** The configuration that you have specified is not supported by Windows. When you start the component created with this combination of parameters, a Windows system call incorrectly retrieves the SID (Security IDentifier) of the computer rather than that of the account.
- **Solution:** If you want to use your local account domain as the domain from which users and groups are drawn, you must specify a different user ID for the service user ID.

  For example, if you specify `NTSecurityDomainName` on the **-d** flag on the **mqsicreatebroker** command as `NYBROKER` on local system \\NYBROKER, the service user ID on the **-i** flag on the **mqsicreatebroker** command cannot be `nybroker`.

  You must delete your User Name Server and re-create it using another user ID that has the correct authorizations.

You have changed your logon password and cannot start your Configuration Manager or broker

- **Scenario:** You have changed your logon password and now you cannot start your Configuration Manager or broker. When you start the Configuration Manager or broker, the following error message is generated:

```
BIP8026E: Unable to start the component.
The component could not be started using the service userid that was supplied when the
component was created. Ensure that the service userid and password are still valid.
Ensure that the service userid has permission to access all of the products directories,
specifically the 'bin' and 'log' directories.  Check for system messages (on Windows NT
this would be the application event log).
```

- **Solution:** You need to change properties of your broker, Configuration Manager, and possibly DB2 services. To do this:
  1. Change your broker using the command:
     ```
     mqsichangebroker brokername -i ServiceUserID -a ServicePassword -p DataSourcePassword
     ```
     For example, to change your logon password to `user1pwd` for user ID `user1` on the broker called `WBRK_BROKER`, use the command:
     ```
     mqsichangebroker WBRK_BROKER -i user1 -a user1pwd -p user1pwd
     ```
  2. Change your Configuration Manager using the command:
     ```
     mqsichangeconfigmgr  -i ServiceUserID -a ServicePassword -p DataBasePassword
     ```
     For example, to change your logon password to `user1pwd` for user ID `user1` on the Configuration Manager, use the command:
     ```
     mqsichangeconfigmgr -i user1 -a user1pwd -p user1pwd
     ```
  3. Change the logon properties on your DB2 services, as follows:
     a. Click **Start** → **Settings** → **Control Panel** → **Administrative Tools** → **Services**
     b. Right-click **DB2 services**, click **Properties**, and click the **Log On** tab
     c. Change your account and password to reflect the database passwords used in the commands above.
     d. Stop and start your services.
     e. Restart your computer.
  4. Restart your components.

Your components should now start.

MQIsdp client connection is refused by broker

- **Scenario:** When a new MQIsdp client attempts to connect to the broker, its connection is refused.
- **Explanation:** MQIsdp Client ID fields must be unique. If a client sends a CONN packet containing the same Client ID as a currently-connected client, the behavior is undefined. It is your responsibility to ensure that Client IDs are unique.

You get abend code 047 and a diagnostic message

- **Scenario:** On z/OS, your broker gets abend code 047 when you try to start it, and the diagnostic message is written to the SDSF SYSLOG:

```
IEA995I SYMPTOM DUMP OUTPUT 463
SYSTEM COMPLETION CODE=047
 TIME=10.53.47  SEQ=00419  CPU=0000  ASID=008E
 PSW AT TIME OF ERROR  078D0000   98D09E52  ILC 2  INTC 6B
   ACTIVE LOAD MODULE           ADDRESS=18D08828  OFFSET=0000162A
   NAME=SPECIALNAME
        61819987 968995A2 A3618499 89A58599 */argoinst/driver*
        F1F46D82 96858261 A4A29961 93979761 *14_boeb/usr/lpp/*
        A6949889 61828995 61828997 89948189 *wmqi/bin/bipimai*
        95                                  *n              *
   DATA AT PSW  18D09E4C - 58109948  0A6B5820  B8E95020
   GPR  0-3  00000000  0000003C  00000000  00000000
```

```
   GPR  4-7  18D10300  18D115F0  00000013  00000004
   GPR  8-11 18D111CF  18D101D0  18D0BBBE  18D0ABBF
   GPR 12-15 98D09BC0  18D101D0  98D09E22  00000000
 END OF SYMPTOM DUMP
```

- **Explanation:** System completion code 047 means that an unauthorized program issued a restricted Supervisor Call (SVC) instruction. The diagnostic message also indicates that the program in error was bipimain.
- **Solution:** When installing WebSphere Message Broker, you must issue the command **extattr +a bipimain** from the bin directory of the installation path to give program bipimain APF authorization.

You get abend code 047 in bin/bipimain

- **Scenario:** You get abend code 047 in bin/bipimain.
- **Explanation:** If you get an 047 abend in bin/bipimain, check the APF authorization.

  If your broker fails to start, submit a job to execute the broker procedure. For example:

  ```
  //MQP1BRK  JOB 1,MSGCLASS=H
  //S1  EXEC PROC=MQP1BRK
  ```

  If this works, the started task user ID is probably unable to access resources.
- **Solution:** To help identify the cause of the problem, carry out the following procedure:
  1. Submit the CVP job to run with the user ID of the broker started task, and resolve any problems.
  2. Submit the CVP job using the started task user ID. You can do this by temporarily putting `user=xxxxx,password=yyyy` on the job card or by using a surrogate facility of the ESM.
  3. To set up the surrogate facility with RACF, create a SURROGAT class profile and give access to either the user only or a group only. For example, use the following RACF commands:

     ```
     RDEFINE SURROGAT MQP1BRK.SUBMIT OWNER(mqp1BRK) AUDIT(ALL(READ))
     PE MQP1BRK.SUBMIT CLASS(SURROGAT) ID(myid) ACC(READ)
     ```

     This allows `myid` to put `USER=MQP1BRK` in the job card of a job and the job runs under the MQP1BRK user ID.

Error message BIP2048E is issued on z/OS when your broker fails to start

- **Scenario:** The following message is written to the SDSF SYSLOG on z/OS when your broker fails to start:

```
+(MA00BRK) 0 BIP2048E: An Exception was caught while issuing database SQL command connect.
+(MA00BRK) 0 BIP2321E: Database error: ODBC return code '-1'.
+(MA00BRK) 0 BIP2322E: Database error: SQL State '58004'; Native Error Code '-99999'
```

- **Explanation:** The SQL State value of 58004 reveals that a system error occurred. This might be due to a DB2 authorization problem.
- **Solution:** The ID under which the broker runs (the started task ID) needs the following DB2 authorizations to be granted:
  - DELETE, INSERT, SELECT and UPDATE authorization on the broker database tables
  - EXECUTE authorization on plan DSNACLI
  - SELECT authorization on table SYSIBM.SYSSYNONYMS
  - SELECT authorization on table SYSIBM.SYSTABLES
  - SELECT authorization on table SYSIBM.SYSDATABASE

Ensure the broker user ID (or the group of which it is a member) has been granted the necessary authorizations.

If the started task ID does not have execute authorization on the DSNACLI plan, you get the errors in the scenario. If ODBC tracing is turned on, you also see the following entries in the traceodbc file for the broker:

```
SQLAllocConnect( hEnv=1, phDbc=&1c6bacec )
SQLAllocConnect( )
    ---> SQL_ERROR
SQLError( hEnv=1, hDbc=0, hStmt=0, pszSqlState=&19ca445c, pfNativeError=&19ca4458,
pszErrorMsg=&1c3d9e50, cbErrorMsgMax=1024, pcbErrorMsg=&19ca4444 )
SQLError( pszSqlState="58004", pfNativeError=-99999, pszErrorMsg="{DB2 for OS/390}{ODBC Driver}
SQLSTATE=58004
ERRLOC=2:170:6
RRS "CREATE THREAD" failed using DB2 system:DFL0 and Plan:DSNACLI RC=08 and REASON=00f30034",
pcbErrorMsg=159)
    ---> SQL_SUCCESS
SQLError( hEnv=1, hDbc=0, hStmt=0, pszSqlState=&19ca445c, pfNativeError=&19ca4458,
pszErrorMsg=&1c3d9e50, cbErrorMsgMax=1024, pcbErrorMsg=&19ca4444 )
SQLError( pszSqlState="42505", pfNativeError=-922,  pszErrorMsg="{DB2 for OS/390}{ODBC Driver}
DSNT408I
SQLCODE = -922, ERROR:  AUTHORIZATION FAILURE: PLAN ACCESS ERROR.
REASON 00F30034
DSNT418I
SQLSTATE = 42505 SQLSTATE RETURN CODE   DSNT415I
SQLERRP = DSNAET03 SQL PROCEDURE DETECTING ERROR    ERRLOC=2:170:8", pcbErrorMsg=371 )
    ---> SQL_SUCCESS
```

If the started task ID does not have SELECT authorization on tables SYSIBM.SYSSYNONYMS, SYSIBM.SYSDATABASE and SYSIBM.SYSTABLES, you see the following type of entry in the traceodbc file if ODBC tracing is turned on:

```
SQLTables( hStmt=1, szTableQualifier=Null Pointer, cbTableQualifier=0,
szTableOwner="MA15USR", cbTableOwner=-3, szTableName=Null Pointer,
cbTableName=0, szTableType=Null Pointer, cbTableType=0 )
SQLTables( )
    ---> SQL_ERROR
```

Error messages BIP2604 or BIP2624 are issued when you start a broker or a new message flow

- **Scenario:** The following messages are written to the USS syslog on z/OS when your execution group, or a newly deployed or started message flow, fails to start:

  ```
  (PMQ1BRK.default)[8]BIP2624E: Unable to connect to queue manager 'PMQ5':
          MQCC=2; MQRC=2025; message flow node 'ComIbmMQConnectionManager'
  (PMQ1BRK.default)[8]BIP2604E: Node failed to open WebsphereMQ queue
         'INPUT1' owned by queue manager 'PMQ5': completion code 2; reason code 2025
  ```

- **Explanation:** The WebSphere MQ return code of 2025 indicates that the maximum number of concurrent connections has been exceeded. On z/OS, a possible cause of this is the setting for IDBACK in the WebSphere MQ CSQ6SYSP macro. Refer to the *WebSphere MQ for z/OS System Setup Guide* on the WebSphere MQ library Web page for information about setting this variable.

Error message BIP2604 with return code MQRC_CONNTAG_IN_USE is issued during the start of a message flow on z/OS

- **Scenario:** The following error message BIP2604 with return code `MQRC_CONNTAG_IN_USE` is displayed during the start of a message flow on z/OS. This message is output every 30 minutes:

  ```
  BIP2604E: Node failed to open WebSphere Business Integration queue [queue name]
  owned by queue manager [queue manager name]
  ```

- **Explanation:** On z/OS, WebSphere MQ supports serialized access to shared resources, such as shared queues, through the use of a connection tag (serialization token) when an application connects to a queue manager that participates in a queue sharing group.

  In this case, a message flow node fails to connect to the indicated WebSphere Business Integration queue manager associated with the input queue because the serialization token that it passed is already in use within the queue sharing group.

  This message is for information only. It indicates that serialization is occurring when two or more message flow input nodes try to connect to a queue manager to get messages from a shared queue.
- **Solution:** Check whether another instance of the message flow, or a flow using the same serialization token is already running. If this is the case, no further action is needed. If this is not the case, contact your IBM Support Center.

A device allocation error is issued

- **Scenario:** A device allocation error is issued.
- **Explanation:** A likely cause is that you do not have the correct permissions set on the component HFS, for the started task ID.
- **Solution:** Check the syslog: if the problem is caused by having incorrect permissions set for the started task ID, you often see a RACF authorization failure message, as shown in the following example.

  ```
  ICH408I USER(TASKID1 ) GROUP(TSOUSER ) NAME(FRED (FRED) 959
   /argo/MA11BRK/ENVFILE
   CL(DIRSRCH ) FID(01D7C7E2E3F0F8000F16000000000003)
   INSUFFICIENT AUTHORITY TO LOOKUP
   ACCESS INTENT(--X)  ACCESS ALLOWED(OTHER ---)
  IEE132I START COMMAND DEVICE ALLOCATION ERROR
  IEA989I SLIP TRAP ID=X33E MATCHED.  JOBNAME=*UNAVAIL, ASID=00A8.
  D J,BPXAS
  IEE115I 11.13.04 2001.212 ACTIVITY 601
  ```

  In this example, the started task ID did not have access to the HFS component. The ICH408I message shows:

  – The file that the task was trying to access

  – The user ID that was trying to access the file

  – The permissions the ID was expecting to have (`INTENT` in the message)

  – The permissions the ID actually had (`ALLOWED` in the message)

  You can use this information to correct the permissions and then reissue, in this example, the start broker request. This type of message is produced if the user issuing the command (which might be to start the broker, or to submit JCL to start one of the utility jobs) does not have the correct HFS permissions for the HFS component. Use the ICH408I information to rectify the problem.

  Another possible reason for authorization failures is inconsistencies in the RACF definitions for a user ID in the MVS image and the OMVS segment. You should also check with your system administrator that the RACF ID used on MVS has a corresponding OMVS image created.

Diagnostic message ICH408I is issued on z/OS when your broker fails to start

- **Scenario:** The following diagnostic message is written to the SDSF SYSLOG on z/OS when your broker fails to start:

  ```
   ICH408I USER(MA10USR ) GROUP(TSOUSER ) NAME(OTHER, A N (ANO) 484
     /argo/MA10BRK/ENVFILE
   -                                          --TIMINGS (MINS.)--
  ```

```
  ----PAGING COUNTS---
 -JOBNAME  STEPNAME PROCSTEP    RC   EXCP    CPU    SRB  CLOCK   SERV  PG
    PAGE   SWAP    VIO SWAPS
   CL(DIRSRCH ) FID(01D7D3E2E3F1F9002D08000000000003)
   INSUFFICIENT AUTHORITY TO LOOKUP
   ACCESS INTENT(--X)  ACCESS ALLOWED(OTHER ---)
```

- **Explanation:** The started task ID under which the broker runs needs to be in a RACF or USS group that has rwx permissions on the broker directory. As an example, consider a broker that is created under directory /argo/MA00BRK. It runs under started task ID MA00USR. Issuing `ls -al` from / to find the permission bit settings on /argo reveals:

  ```
  drwxrwx---  5 BPXROOT  ARGOUSR      8192 Jul 30  13:57  argo
  ```

  Issuing `id MA00USR` to find the group membership of started task ID MA00USR reveals:

  ```
  uid=14938(MA00USR) gid=5(TSOUSER) groups=229(ARGOUSR)
  ```

  This means that the started task ID MA00USR potentially has rwx permissions on subdirectories to /argo because these are set for both the user and the group associated with MA00USR. If the permissions are not set correctly, you get the type of diagnostic message shown in the scenario.
- **Solution:** Make sure that the started task ID under which the broker runs is in an RACF or USS group that has rwx permissions on the broker directory.

When you start the broker through the Dataflowengine, it cycles continually

- **Scenario:** When you start the broker through the Dataflowengine, it continually cycles, starts and stops, and the following errors appear in the log:
  - BIP2801E: `Unable to load implementation file '/opt/IBM/DistHub/v2/lib/libdhbNBIO.so', rc=The file access permissions do not allow the specified action.`
  - BIP2110E: `Message broker internal program error.`
- **Explanation:** The permissions on /opt/IBM are 700, meaning that the Broker ID cannot read the disthub files.
- **Solution:** Ensure that permissions on /opt/IBM are set to 755, which is rwxr-xr-x.

The create command fails and error message BIP8022E is issued

- **Scenario:** Error code BIP8022E is displayed when you use the **mqsicreateconfigmgr**, **mqsicreateusernameserver**, or **mqsicreatebroker** command on Windows 2000 or Windows XP, even if the supplied user name and password are correct.
- **Explanation:** The Microsoft component ″Shared File and Printer Services″ is required.
- **Solution:** Rectify this error by installing the ″Share File and Printer for Microsoft network″ service on the Windows platform.

Broker terminates with an abend on startup on AIX

- **Scenario:** The broker terminates with an abend on startup. You see a BIP2228E message saying that the /usr/java131 directory cannot be found and a number of abend files are generated.
- **Explanation:** The Java Runtime Environment (JRE) is a prerequisite on AIX and must be installed in a directory named /usr/java131 in order to start a broker.
- **Solution:** Make sure that you install the JRE in the /usr/java131 directory before starting a broker. Alternatively, create a symbolic link between the /usr/java131 directory and the directory where the JRE was installed.

Error code BIP0889E is issued when you connect to the Configuration Manager

- **Scenario:** Error code BIP0889E is displayed when you connect to the Configuration Manager from the tooling.
- **Explanation:** The Windows service ″Client for Microsoft network″ is required.
- **Solution:** Rectify this error by installing the ’Client for Microsoft network’ service on the Windows platform.

Connection to the Configuration Manager is slow

- **Scenario:** Connection to the Configuration Manager is slow and you are unable to perform other actions in the tooling when performing the following operations:
  - connecting to a domain
  - creating a new broker
  - creating a new execution group
  - deploying a broker archive file, using the **Deploy a BAR File** wizard
  - opening the Broker Topology editor
  - opening the Topics Hierarchy editor
  - opening the Subscriptions Query editor
  - opening the Event Log editor
- **Solution:** Click **Cancel** on the progress monitor dialog box to cancel the in-progress connection. Connection to the Configuration Manager is cancelled, allowing you to perform other actions in the tooling. You can then resubmit the cancelled operation.

You experience problems connecting to the Configuration Manager queue manager

- **Scenario:** You experience problems when connecting to the Configuration Manager queue manager from the tooling.
- **Explanation:** The Message Brokers Toolkit sends the user and domain name through to the Configuration Manager queue manager regardless of the domain awareness setting on the Configuration Manager. This can cause problems connecting to the queue manager because of the security required to connect, put, or get messages.
- **Solution:** To turn off the domain awareness on the Message Brokers Toolkit, run the Message Brokers Toolkit in the following way:
  1. Change to the <install_wbimb_dir>\eclipse directory.
  2. Run the Message Brokers Toolkit using the command **mqsistudio -vmargs -DDomainAware=0**.

  Alternatively, modify the shortcut that runs the Message Brokers Toolkit and add on **-vmargs -DDomainAware=0**.

For more information, see Considering security for the workbench.

Resolving problems when importing or migrating resources

This topic contains advice for dealing with some common problems that can arise when importing or migrating resources:

- “The mqsimigratemsgflows command gives unexpected results” on page 28
- “The mqsimigratemsgsets command gives unexpected results” on page 28

- “Message flows referring to a migrated user-defined node have connection errors”
- “Message flows that have been migrated from Version 2.1 cannot resolve subflows” on page 29
- “A broker has been migrated from Version 2.1 to Version 6.0 and some message flows are now stopped” on page 29
- “Error messages are issued after a Version 2.1 broker is migrated to Version 6.0” on page 29
- “After migration, message flows cannot locate a user-defined node” on page 29
- “Deployment of message flows fails when you migrate them from Version 2.1 to Version 6.0” on page 29
- “You get import problems with imported message sets” on page 30
- “File > Import gives you only the option to import a zipped file inside an existing project” on page 30
- “Source directory name contains spaces” on page 30
- “Importing database files after the first import results in strange behavior” on page 30
- “Migration to the Version 6.0 Configuration Manager repository is unsuccessful” on page 31
- “Warnings are displayed for message flows that you have imported to Version 6.0 from Version 2.1” on page 31
- “You migrated a Configuration Manager to another computer, redefined a broker in the new Configuration Manager and used the Configuration Manager Proxy to modify the broker reference’s UUID.” on page 32
- “You encounter problems when you try to connect to a migrated Configuration Manager from the Version 6.0 Message Brokers Toolkit” on page 32

The mqsimigratemsgflows command gives unexpected results

- **Scenario:** You have imported your Version 2.1 or Version 5.0 message flows into the workbench using the **mqsimigratemsgflows** command, but the results are not quite what you were expecting.
- **Explanation:** The **mqsimigratemsgflows** command creates a report called mqsimigratemsgflows.report.txt in the directory from which you ran the command (typically the /eclipse directory). Read the report for details of the actions taken by the command.

The mqsimigratemsgsets command gives unexpected results

- **Scenario:** You have imported your Version 2.1 or Version 5.0 message sets into the workbench using the **mqsimigratemsgsets** command, but the results are not quite what you were expecting.
- **Explanation:** The **mqsimigratemsgsets**command creates a report called mqsimigratemsgsets.report.txt in the directory from which you ran the command (typically the /eclipse directory). Read the report for details of the actions taken by the command.

Message flows referring to a migrated user-defined node have connection errors

- **Scenario:** After migration, all message flows referring to a migrated user-defined node have errors indicating that connections cannot be made.
- **Explanation:** One possible cause is that the original user-defined node had space characters as part of one or more terminal names. The spaces are rendered as ’X20’ erroneously.

- **Solution:** You can correct this by editing the user-defined node .msgnodefile. After migration the .msgnode file is in the same project where flows were migrated. Open the editor and correct the terminal name. Ensure that the name is exactly as the broker node implementation expects.

Message flows that have been migrated from Version 2.1 cannot resolve subflows

- **Scenario:** A set of message flows has been migrated from Version 2.1 to Version 6.0 and an error message is issued indicating that: `Subflow` *`message flow name`* `cannot be located`.
- **Explanation:** The subflow in the message flow has not been migrated correctly. The message flow was exported from Version 2.1 without the subflow or it was exported into a different file to the subflow.
- **Solution:** Export everything from Version 2.1 into one large export file, then migrate the export file to Version 6.0. Alternatively, if the subflow has been imported but the message flow still shows the error, right-click the node that is causing the error and click **Locate subflow**.

A broker has been migrated from Version 2.1 to Version 6.0 and some message flows are now stopped

- **Scenario:** A broker has been migrated from Version 2.1 to Version 6.0 and some message flows are now stopped
- **Explanation:** This error can be caused by invalid ESQL in the deployed message flows that was not detected on the Version 2.1 broker.
- **Solution:** Remove the affected message flows from the execution group, then redeploy the message flows. This provides more information about the error so that you can resolve it.

Error messages are issued after a Version 2.1 broker is migrated to Version 6.0

- **Scenario:** The following error messages are issued after a Version 2.1 broker is migrated to Version 6.0: BIP2238, BIP2239, BIP2280, BIP2281, and BIP2308.
- **Explanation:** These error messages are issued when there are clashes between Version 6.0 and Version 2.1 lil files.
- **Solution:** These error messages do not cause any problems and you can ignore them. You can remove them by using the **-l** parameter of the **mqsichangebroker** command to change the lil path that is used by the broker. If all the Version 2.1 lils are removed from the lil path, the errors will not appear.

After migration, message flows cannot locate a user-defined node

- **Scenario:** After migration, message flows cannot locate a user-defined node.
- **Explanation:** One possible cause is that the flows do not internally have the correct reference to a user-defined node.
- **Solution:** Invoke the **Locate subflow** pop-up menu from the node that cannot be located. Using the Browse dialog, locate the user-defined node (which is in the same project as migrated flows). The message flow should now link to the user-defined node correctly and the task list entry will be removed when the flow is saved.

Deployment of message flows fails when you migrate them from Version 2.1 to Version 6.0

- **Scenario:** After migration from Version 2.1 to Version 6.0, the deployment of message flows fails and a BIP2493 error message is displayed.

- **Solution:** Modify the ESQL so that it references a list rather than a scalar value. A list is denoted by an empty set of square brackets ("[ ]"), as in the example:

  ```
  InputRoot.XMl.Field1[]
  ```

You get import problems with imported message sets

- **Scenario:** Your message set worked with Version 2.1, but when you imported it into WebSphere Message Broker Version 6.0, you get import problems or task list errors.
- **Explanation:** Within Version 2.1, message sets were checked at several different levels. A message set was not completely checked until it was deployed to a broker. If your Version 2.1 message set was never deployed to a broker, the checks will not have been completed, and as a result you might not have been informed about problems with the message set. WebSphere Message Broker Version 6.0 ensures that message sets are completely checked whenever they are saved. This ensures that you are informed about problems with the message sets as soon as possible. As a result, Version 2.1 message sets might contain errors that are reported as soon as you import the message sets into WebSphere Message Broker Version 6.0.
- **Solution:** You should fix the problems using either Version 2.1, remembering to deploy the message set to be sure that the checks are complete, or WebSphere Message Broker Version 6.0.

File > Import gives you only the option to import a zipped file inside an existing project

- **Scenario:** You have a zipped file that contains message set projects and message flow projects. When you click **File** → **Import**, you only have the option to import the zipped file inside an existing project, but you want to re-create the message set projects and message flow projects.
- **Solution:** When you export and import files, do not export or import the root directory, which gets created anyway because of the project file. When you export your message flow and message set projects:
  1. Click **Create only selected directories**.
  2. Clear the project root folder.
  3. Select the files and subdirectories as required. The project root folder is selected, but gray.

  Then, when you import the zipped file:
  1. Deselect the root (/) folder.
  2. Select the files and subfolders as required.

  The project root folder is selected, but gray.

Source directory name contains spaces

- **Scenario:** You want to use the **mqsimigratemsgflows**or **mqsimigratemsgsets** command, but your source directory contains embedded spaces.
- **Solution:** Put the directory name in double quotes. For example:

  ```
  mqsimigratemsgflows -p MYFLOW01 -d "E:\Program Files\My message flows"
  ```

Importing database files after the first import results in strange behavior

- **Scenario:** Importing database files after the first import results in strange behavior. If you re-import a table, the mapping file displays a broken reference to any mappings that were created below the table. If you import additional tables for mapping, you get unresolvable table reference warnings in the task list for references to previously imported tables. When you use content assist in the ESQL editor, there are suggestions for only some of the imported tables.

- **Explanation:** Performing multiple imports on the same database generates new schema files for the connection (`.conxmi`), database (`.dbxmi`), schema (`.schxmi`) and selected tables (`.tblxmi`). Your schema files might be out of sync if you are trying to import into the same location. Because new files are created, links to previously imported table files will be broken.
- **Solution:** You can either create a new location for the import, or you can delete all the related schemas for the database and import into the same location. Any mappings you have will be preserved as long as the tables are selected again.

The import action in the **Add Database Table Schemas** dialog is not identical to the action in the Data perspective. Using these two import actions interchangeably to the same target folder causes unexpected behavior.

Migration to the Version 6.0 Configuration Manager repository is unsuccessful

- **Scenario:** Migration to the Version 6.0 Configuration Manager repository is unsuccessful.
- **Explanation:** If migration to the internal repository cannot be completed successfully, the internal repository remains empty but it is primed with an empty topology. The Configuration Manager attempts to start up after a failed migration attempt and reports any problems in the system log.
- **Solution:** To migrate DB2-based configuration data, take the following steps:
  1. Stop the Configuration Manager.
  2. Delete the whole Configuration Manager using the **mqsideleteconfigmgr** command, specifying the **-n** parameter, which causes the configuration repository to be deleted.
  3. Re-create the Configuration Manager using the **mqsicreateconfigmgr** command, specifying the DB2 database for migration.

Warnings are displayed for message flows that you have imported to Version 6.0 from Version 2.1

- **Scenario:** Warning messages are displayed for message flows that you have imported to Version 6.0 from Version 2.1, such as `Unresolvable message field reference Body.RootTag.ID`.
- **Explanation:**Version 6.0 warns of any field reference that it cannot resolve. These warnings do not cause any problems with deploying or running the message flows.
- **Solution:** You can remove a large number of warning messages by performing the following steps:
  1. Reference any MRM message sets in the message flow projects **Property** pane. This removes any warnings for messages defined in the message sets.
  2. Run the rdb definition file wizard to define all the databases that are used by the message flows.

You migrated a Configuration Manager to another computer, redefined a broker in the new Configuration Manager and used the Configuration Manager Proxy to modify the broker reference's UUID.

- **Scenario:** During the migration of a Configuration Manager from one computer to another, you attempted to redefine a broker in the new Configuration Manager and use the Configuration Manager Proxy to modify the broker reference's unique identifier (UUID).
- **Explanation:** This is not a valid procedure for migration. In order to migrate a Configuration Manager from one machine to another, use one of the following methods:
  - Use the **mqsibackupconfigmgr** command to back up the Configuration Manager on the source computer, and use the **mqsirestoreconfigmgr** command on the target computer to move the Configuration Manager.
  - Specify the source computer's DB2 data source on the **mqsicreateconfigmgr** command.
- **Solution:** Delete the Configuration Manager on the target computer and follow the instructions given in the appropriate topic:
  - Migrating a Configuration Manager from Version 2.1
  - Migrating a Configuration Manager from Version 5.0

You encounter problems when you try to connect to a migrated Configuration Manager from the Version 6.0 Message Brokers Toolkit

- **Scenario:** You encounter problems when you try to connect to the migrated Configuration Manager from the Version 6.0 Message Brokers Toolkit.
- **Explanation:** This problem can occur if you have configured MCA users on the WebSphere MQ channels to the Configuration Manager.
- **Solution:** Configure access control lists (ACLs) for users who are running the Version 6.0 Message Brokers Toolkit. Follow the instructions in Considering security for the workbench.

Resolving problems when stopping resources

This topic contains advice for dealing with some common problems that can arise when stopping resources:

- "You cannot stop the broker"
- "You cannot stop the broker queue manager" on page 33
- "Execution group terminates abnormally" on page 33

You cannot stop the broker

- **Scenario:** You issue the **mqsistop** command to stop the broker, but the system freezes and never actually stops any of the execution groups.
- **Explanation:** One possible cause is that you or someone else is debugging a flow and it is currently stopped at a breakpoint. WebSphere Message Broker regards this as a message in flight situation, and refuses to stop the broker through the normal command.
- **Solution:** Click **Stop debugging** from the workbench (or ask the person doing the debugging to do this). After that operation has completed, the broker stops.

If it is not possible to stop debugging, end all execution group processes associated with that broker to allow the broker to stop. However, your messages are backed out and you still have to click **Stop debugging** after the broker restarts.

You cannot stop the broker queue manager

- **Scenario:** You are trying to use the WebSphere MQ **endmqm** command to stop a broker queue manager on a distributed platform, but it does not stop.
- **Explanation:** In certain circumstances, attempting to stop a broker queue manager does not cause the queue manager to stop. This can happen if you have configured any message flows with multiple threads (you have set the message flow property Additional Instances to a number greater than zero).
- **Solution:** If you want to stop the broker's queue manager, stop the broker using the **mqsistop** command specifying the **-q** flag. (The **-q** flag is not available on z/OS.) This issues the WebSphere MQ **endmqm** command on your behalf in a controlled fashion that shuts down the broker and the queue manager cleanly.

Execution group terminates abnormally

- **Scenario:** Your execution group processes terminate abnormally.
- **Explanation:** When execution group processes terminate abnormally, they are automatically restarted by the bipbroker process. If an execution group process fails, it is restarted three times during each five-minute interval. The first five-minute interval begins when the execution group is first started. The broker database table BROKERAAEG contains a column labeled RetryInterval. This column defines the restart time in minutes. Each row in this table represents an execution group. The row is populated on the first deploy of an execution group and RetryInterval defaults to 5.

  If you remove the execution group from the broker configuration, deploy the broker configuration, and then later add the execution group and redeploy the broker configuration. The row is re-created and RetryInterval is set to its default value of 5.
- **Solution:** To change the default value:
  1. Stop the broker.
  2. Change the RetryInterval's value in the database table.
  3. Restart the broker.

Resolving problems when deleting resources

This topic contains advice for dealing with some common problems that can arise when deleting resources:

Problems when deleting a broker:

- "You cannot delete your broker after deleting your broker database" on page 34
- "The workbench is deploying to a deleted broker" on page 34
- "You cannot delete a broker following a disk failure, or the Configuration Manager contains references to a deleted broker" on page 35
- "The **mqsideletebroker** command hangs if the DB2 profile has not been run" on page 36

Problems when deleting other resources:

- “Error message BIP8052E is issued when you delete the Configuration Manager” on page 36
- “You cannot delete a project from your workspace” on page 36
- “You want to remove a domain from the Domains navigator” on page 36

You cannot delete your broker after deleting your broker database

- **Scenario:** You deleted your broker database, and now you cannot delete your broker, and get error message BIP8040W.
- **Explanation:** The **mqsideletebroker** command checks for the broker database tables and raises this error because the database is not there.
- **Solution:** You can work around this problem by creating a dummy database with the same name as the database that you deleted. You must also re-create the ODBC connection. Reissue **mqsideletebroker**, and then delete the dummy database.

The workbench is deploying to a deleted broker

- **Scenario:** You have tried to delete your broker using the **mqsideletebroker**command, and deleted it from the topology, but the workbench is still deploying to the deleted broker.
- **Explanation:** This problem occurs because you have either performed an incorrect sequence of actions when you deleted the broker, or you have re-created a broker of the same name before the previous deletion has completed.
- **Solution:**These steps explain how to delete a broker correctly. If a step completed successfully when you previously tried to delete your broker, you do not need to repeat it.
  1. Delete the broker from the topology.
  2. In the Event Log editor, look in a response message from the Configuration Manager that states that the broker has updated its configuration and all references to the broker have been removed from the configuration repository. If you see an error, correct the problem that is reported and perform a delta deploy of the topology, which asks the broker to tidy up its resources.
  3. After the broker resources have been tidied and the broker has been removed from the Configuration Manager repository, stop the broker using the **mqsistop** command.
  4. Delete the broker using the **mqsideletebroker** command.

If the deploy step fails because the broker has already been deleted using the **mqsideletebroker** command, or it is otherwise unable to respond, remove completely all references to the broker from the Configuration Manager repository. You can do this by removing the broker from the topology using the Message Brokers Toolkit. Ensure that the broker’s SYSTEM.BROKER.ADMIN.QUEUE queue has been deleted and initiate a further topology deploy.

Alternatively, you can use the Configuration Manager Proxy API ″TopologyProxy.removeDeletedBroker(String)″ to programmatically purge all references to the broker:

1. Start the Configuration Manager Proxy API Exerciser sample (which is available on the **Start** menu in Windows).
2. Connect to the Configuration Manager.

3. Right-click the topology object and click **Remove references to a previously deleted broker**.

The broker should now have disappeared from the workbench.

Do not try to re-create a broker using the **mqsicreatebroker**command until you are confident that it has been removed from the workbench. Similarly, do not reuse broker or queue manager names until you are confident that they are not in use elsewhere.

You cannot delete a broker following a disk failure, or the Configuration Manager contains references to a deleted broker

- **Scenario:** You have had a complete disk failure on your broker system. You have decided that you no longer need the broker that was on that system, so now you want to delete this broker from the topology.
- **Explanation:** Under normal circumstances, the broker is deleted from the topology, and the topology is redeployed before the broker is deleted using the **mqsideletebroker**command. Even if the broker was deleted using the **mqsideletebroker** command before the topology was updated, a redeploy of the topology detects that the broker has been deleted. This happens because the broker's control queues no longer exist on the broker's queue manager and the redeploy can detect the missing queues.

  However, when a system crashes, it is impossible to start the remote queue manager, or even the channels connecting to it. All deploys to the crashed system result in messages being left on the transmission queue for the remote queue manager. So, from the perspective of the workbench, the symptom of the failing deploy is a remote queue manager that is not responding.
- **Solution:** To force the deletion of the lost broker from the topology:
  1. Create a new queue manager with the same name as that used by the broker that you have lost.
  2. Start this new queue manager.
  3. Connect the new queue manager to the Configuration Manager's queue manager, that is, create its transmission queue and its sender and receiver channels.
  4. Start these channels.
  5. Remove the broker from the Topology editor and save the topology. The Configuration Manager detects that the broker's queues no longer exist and it removes the broker from the configuration repository.
- **Alternative solution:** References to deleted brokers can be removed programmatically from the Configuration Manager's repository using the Configuration Manager Proxy API. Use the API ″TopologyProxy.removeDeletedBroker(String)″ to do this. You can also do this by using the Configuration Manager Proxy API Exerciser sample:
  1. Start the Configuration Manager Proxy API Exerciser sample (which is available on the **Start** menu in Windows).
  2. Connect to the Configuration Manager.
  3. Click the PubSubTopology object to select it, then right-click and click **Remove references to a previously deleted broker**.
  4. Enter the name of the broker that you wish to delete.

CAUTION:
Do not remove a working broker using this facility. By removing references to a broker from the Configuration Manager repository, you will make it unmanageable.

The mqsideletebroker command hangs if the DB2 profile has not been run

- **Scenario:** The **mqsideletebroker** command hangs if the DB2 profile has not been run.
- **Solution:** Run the DB2 profile before re-issuing the **mqsideletebroker** command.

Error message BIP8052E is issued when you delete the Configuration Manager

- **Scenario:** Error message BIP8052E is displayed when you delete the Configuration Manager using the **mqsideleteconfigmgr**.
- **Explanation:** Error message BIP8052E indicates that the queue SYSTEM.BROKER.SECURITY.REPLY, one of the WebSphere MQ queues created specifically for the Configuration Manager's use, cannot be deleted and the action fails. This is probably because the queue is currently in use.
- **Solution:** Make sure that your Configuration Manager is stopped, then retry the **mqsideleteconfigmgr** command after a short interval.

You cannot delete a project from your workspace

- **Scenario:** You cannot delete a project from your workspace. You get error messages indicating that the containing directory cannot be deleted, or the project file is missing.
- **Explanation:** If you attempt to delete a project, and the directory containing the project is in use, or you have any files that are contained within the project open externally to the workbench, some of the resources in the project are not deleted, but others, including the project file, might be deleted.
- **Solution:** Before attempting to delete a project, make sure that you have no files open externally, and that you do not have an open command prompt located in the directory. To recover from this problem, manually delete any remaining files and directories from your workspace directory, then click **Delete** from the project in the workbench.

You want to remove a domain from the Domains navigator

- **Scenario:** You want to remove a domain from the Domains navigator.
- **Explanation:** You cannot remove a domain from the Domains navigator.
- **Solution:** Ensure that you have disconnected from the domain, and then delete the associated domain connection from the Broker Administration perspective. The domain targeted by this domain connection is then removed from the Domains navigator.

Resolving problems when developing message flows

This topic contains advice for dealing with some common problems that can arise when developing message flows:

Migration

- "Your migrated message flow project contains .msgnode files" on page 38
- "You do not know whether you need to migrate the IBM primitives" on page 38

- “An error message is issued when you save a message flow that contains MQe nodes” on page 38

ESQL

- “A ″routine not defined″ error message is issued in ESQL when you move a routine” on page 38
- “The product hangs when pasting ESQL statements from Adobe Acrobat” on page 39
- “You do not know how message flows handle the code page of ESQL files” on page 39
- “You do not know the naming restrictions for ESQL procedures and functions” on page 39
- “Error message BIP5431E is issued and the broker fails” on page 39
- “You have problems calling Java from ESQL” on page 39
- “Your format expression contains non-valid characters” on page 39
- “Your input string does not match the given format expression” on page 40
- “Your format expression is not applicable” on page 40

Mappings and message references

- “Resources referenced by the mapping file cannot be resolved” on page 40
- “You get errors when importing table schemas into the Message Flow Mapping editor” on page 40
- “You get warnings or errors for message references” on page 41

Execution

- “Messages are directed to the failure terminal of an MQInput node” on page 41
- “Error message BIP2211E is issued by the MQInput node in z/OS” on page 41
- “Messages enter the message flow but do not exit” on page 42
- “Your execution group is not reading messages from the input queues” on page 43
- “DataFlowEngine terminates while processing messages” on page 44
- “Your execution group hangs or terminates with a core dump” on page 44
- “″Style sheet could not be processed″ errors are issued and your XMLTransformation node is not working after deployment” on page 45
- “Output messages are not sent to expected destinations” on page 45

Trace

- “You cannot determine which node is being referenced in your trace file” on page 45
- “You cannot see any alerts when you change user trace” on page 46

Appearance

- “Task list does not update when you make corrections to your files” on page 46
- “You rename a flow that contains errors, but the task list entries remain” on page 46

Other

Your migrated message flow project contains .msgnode files

- **Scenario:** After migrating message flows, you see some `.msgnode` files in the root directory of the Eclipse project, and do not know what to do with them.
- **Explanation:** This happens because one or more Version 2.1 export files contained user-defined nodes (UDNs). It is likely that the flows that you just migrated are using these UDNs.
- **Solution:** If you are the author of these UDNs:
  1. Create a UDN project.
  2. Move the files into that project.
  3. Complete the development process for the UDNs.

  If you are not the author of the UDNs, you can delete these files. The author, or vendor, should provide the 5.0 version of these UDNs in the form of an Eclipse feature and plug-in. The plug-in should contain icons, translation, palette definition, infopops, help, and so on.

You do not know whether you need to migrate the IBM primitives

- **Scenario:** You do not know whether you need to migrate the message flow IBM primitives.
- **Explanation:** You do not need to migrate these primitives. The Version 2.1 primitives are supplied with WebSphere Message Broker Version 6.0. The Version 2.1 export file does not contain IBM primitives.

An error message is issued when you save a message flow that contains MQe nodes

- **Scenario:** An error message is issued when you save a message flow that contains MQe nodes.
- **Explanation:** The use of message flows that contain MQeInput and MQeOutput nodes in WebSphere Message Broker Version 6.0 is deprecated. It is still possible to use them to deploy from Version 6.0 to previous versions.
- **Solution:** Override the error and save. When you deploy the message flow to a previous version broker, it works as before. When you deploy the message flow to a Version 6.0 broker, create a suitable configuration to deal with it, as described in Configuring after MQe flow deployment.

A ″routine not defined″ error message is issued in ESQL when you move a routine

- **Scenario:** A `Routine not defined` error message is displayed in ESQL when you move a routine from one schema to another.
- **Explanation:** If a routine that was referenced by code in one schema is moved to another schema, where it is still visible, a false error is generated stating that the routine cannot be resolved.

- **Solution:**Clean the project by clicking **Project** → **Clean**.

The product hangs when pasting ESQL statements from Adobe Acrobat

- **Scenario:** When you copy and paste certain ESQL statements from Adobe Acrobat into the ESQL editor, WebSphere Message Broker stops responding.
- **Explanation:** This problem occurs when pasting text from Adobe Acrobat into the ESQL editor and also into the Java editor.
- **Solution:** To work around this problem, either manually enter the text, or copy and paste to a text editor like Notepad and perform another copy and paste action from there.

You do not know how message flows handle the code page of ESQL files

- **Scenario:** You do not know how message flows handle the code page of ESQL files.
- **Solution:** The code page of an ESQL file is the code page of the workbench on which the file is created. You must deploy a message flow using an ESQL file on a workbench with the same code page setting as the ESQL file. When multiple ESQL files are involved in a single compiled message flow (`.cmf`) file, all these ESQL files must be in the same code page.

  See Editor preferences and localized settings for more information.

You do not know the naming restrictions for ESQL procedures and functions

- **Scenario:** You do not know the restrictions for choosing names for ESQL modules or schema scope ESQL and mapping procedures and functions.
- **Solution:** Module and schema scope procedures and functions cannot have names starting with `IBM_WBIMB_`. `IBM_` is reserved for IBM use, and `IBM_WBIMB_` is reserved for WebSphere Message Broker.

Error message BIP5431E is issued and the broker fails

- **Scenario:** Error message BIP5431E is displayed and the broker fails.
- **Explanation:** When setting output message properties, you have specified an incorrect physical format name for the Message Format.
- **Solution:** The name that you specify for the physical layer must match the name that you have defined for it. The default physical layer names are `CWF1`, `XML1` and `TDS1`.

You have problems calling Java from ESQL

- **Scenario:** Your Java class files are not being found.
- **Explanation:** When creating the class files, you have not placed them in the correct location within the system CLASSPATH.
- **Solution:** See the CREATE PROCEDURE statement for further information.

Your format expression contains non-valid characters

- **Scenario:** Your format expression contains an unrecognized character for the given conversion.
- **Explanation:** Your format expression for a numerical conversion was used to convert to or from a *DATE*, *TIME*, *TIMESTAMP*, *GMTTIME* or *GMTTIMESTAMP* variable. Another possible explanation is that your format expression for a DateTime conversion was used to convert to or from an *INTEGER*, *DECIMAL* or *FLOAT* variable.

- **Solution:** You should replace the format expression with one of the applicable type. For more information about valid data types and expressions, see ESQL reference.

Your input string does not match the given format expression

- **Scenario:** You have used an input string that does not match the given format expression.
- **Explanation:** Your given format expression contains data which does not match the current element of the format expression.
- **Solution:** You should either rewrite the format expression to match the input data, or modify the input data to match the format expression. For more information about valid data types and expressions, see ESQL reference.

Your format expression is not applicable

- **Scenario:** You have used a format expression when it is not applicable, for example when converting from decimal to integer.
- **Explanation:** The use of format expressions is limited to casting between datetime and string values or numerical and string values. Your format expression cannot be applied in this case.
- **Solution:** You should either remove the FORMAT clause, or change the parameters. For more information about valid data types and expressions, see ESQL reference.

Resources referenced by the mapping file cannot be resolved

- **Scenario:** You have imported some message flows into the workbench that contain mappings. You have an error showing that the resources referenced by the mapping file cannot be resolved.
- **Explanation:** Mappings can use resources that exist in other projects. For example, a mapping reference to a message set would exist in a different project. If the reference cannot be resolved, it probably means that the reference to the other project has been lost.
- **Solution:** To reference a project:
  1. Right-click the project with the error, and click **Properties**.
  2. Under Project References, a list of the projects in your workspace is displayed. Select the required projects to reference the resources in these projects.
  3. Click **OK**.

You get errors when importing table schemas into the Message Flow Mapping editor

- **Scenario:** When you try to import and add table schemas in the Message Flow Mapping editor, you encounter errors like:

  ```
  /flow2/schema1/SAMPLE.conxmi cannot be loaded.
  The following error was reported: schema1/SAMPLE.conxmi
  ```
- **Explanation:** This error usually means that you have the same database files under the same Broker Schema name in another project. Because the relative paths are the same, the Message Flow Mapping editor cannot resolve this ambiguity and does not know which table to add.
- **Solution:** Depending on what your intentions were, you can take one of two courses of action:
  - If the table file already exists in the workspace, and this is what you want to use for mapping, reuse the file by selecting the **Add database table schemas from workspace** option in the Add Database Table Schemas dialog.

– If you wanted a different copy of the tables, rename the Broker Schema.

You get warnings or errors for message references

- **Scenario:** You get warnings or errors for message references, and you are sure that your references are correct.

  Note: This will never be the case with messages using the XML parser. For these message references, direct validation is not performed because the references could be used for generic XML.
- **Solution:** To use the validation feature:
  1. Create a reference to the tree and parser in the module main procedure.
  2. Associate the reference to the correlation name, for example InputRoot, Root, or create the OutputRoot.*parser* node, where *parser* is the name of the parser that you want to use.
  3. Pass the reference as a parameter to an ESQL subroutine that identifies the XSD type of the reference.

This practice is beneficial because the passed reference supports content-assistance and validation for ESQL. The message type content properties `open`, or `open defined` are not used in validation, and the assumption is that this property is `closed`.

There is an ESQL editor preference that lets you choose to ignore message reference mismatches, or to have them reported as a warning or an error. By default, this type of problem is reported as a warning, so you can still deploy the message flow.

Messages are directed to the failure terminal of an MQInput node

- **Scenario:** Messages that are received at a message flow are immediately directed to the failure terminal on the MQInput node (if it is connected) or are rolled back.
- **Explanation:** When a message is received by WebSphere MQ, an error is signalled if the following are all true:
  - The MQInput node requests that the message content is converted (the Convert property is set to `yes` on the node).
  - The message consists solely of an MQMD followed by the body of the message.
  - The format as specified in the MQMD is set to MQFMT_NONE.

  This error causes the message to be directed to the failure terminal.
- **Solution:** In general, there is no need to request WebSphere MQ to convert the message content, because WebSphere Message Broker processes messages in any code page and encoding supported by WebSphere MQ. Set the Convert property to `no` to ensure that messages flow from the MQInput node to successive nodes in the message flow.

Error message BIP2211E is issued by the MQInput node in z/OS

- **Scenario:** The following error message is issued by the MQInput node, indicating an invalid attribute:

  ```
  BIP2211: (Invalid configuration message containing attribute value [attribute value]
  which is not valid for target attribute [target attribute name],
  object [object name]; valid values are [valid values])
  ```
- **Explanation:** On z/OS, WebSphere MQ supports serialized access to shared resources, such as shared queues, through the use of a connection tag

(serialization token) when an application connects to the queue manager that participates in a queue sharing group. In this case, an invalid attribute has been specified for the z/OS serialization token.

- **Solution:** Check that the value provided for the z/OS serialization token conforms to the rules as described in the *WebSphere MQ Application Programming Reference* on the WebSphere MQ library Web page.

Messages enter the message flow but do not exit

- **Scenario:** You are sending messages into your message flow, and they are removed from the input queue, but nothing ever appears at the other end of the message flow.
- **Explanation:** There are several possible causes for this. Work through the suggestions given until you can identify the situation that is causing your failure:
  1. Check your message flow in the workbench.

     You might have accidentally connected the MQInput node failure terminal to a successive node instead of the out terminal. The out terminal is the middle terminal of the three. Messages directed to an unconnected out terminal are discarded.
  2. If the out terminal of the MQInput node is correctly connected to a successive node, check in the local error log for the broker for a message indicating that message processing has been terminated due to problems. Additional messages give more detailed information.

     If the failure terminal of the MQInput node has been connected, for example, to an MQOutput node, these messages do not appear.

     Connecting a node to a failure terminal of any node indicates that you have designed the message flow to deal with all error processing. If you connect a failure terminal to an MQOutput node, your message flow ignores any errors that occur.
  3. If the out terminal of the MQInput node is correctly connected to a successive node, and the local error log does not contain any error messages, turn user tracing on for the message flow:
     1. Switch to the Broker Administration perspective.
     2. In the Domain view, select the message flow, and right-click. (You might have to refresh the view by clicking the **Refresh** icon.)
     3. Click **User Trace** → **Normal**.

     This action produces a user trace entry from every node that the message visits, and only those nodes.

     On distributed platforms, you can retrieve the trace entries using the **mqsireadlog** command, format them using the **mqsiformatlog** command, and view the formatted records to check the path of the message through the message flow.

     For z/OS, edit and submit the job `BIPJLOG` in the `COMPONENTPDS` to execute the **mqsireadlog** and **mqsiformatlog** commands to process traces.
  4. If the user trace shows that the message is not taking the path through the message flow that you expect, increase the user trace level to Debug by selecting the message flow, right-clicking on it, and clicking **User Trace** → **Debug**.

     Send your message into the message flow again. Debug level trace produces much more detail about why the message is taking the route that it is taking, and you can then determine the reasons for the actions taken by the message flow.

Do not forget to turn tracing off when you have solved the problem, or performance will be adversely affected.

5. If the **mqput** command to the output queue defined on the MQOutput node is not successful (for example, the queue is full or put is disabled), the final destination of a message depends on:
   - Whether the failure terminal of the MQOutput node is connected.
   - Whether the message is being processed transactionally (which in turn depends on the transaction mode setting of the MQInput node, the MQOutput node, and the input and output queues).
   - Whether the message is persistent or nonpersistent (transaction mode=automatic only). All messages are treated as persistent if transaction mode=yes, and as nonpersistent if transaction mode=no.

   In general:
   - If a path is not defined for a failure (that is, neither the catch terminal nor the failure terminal of the MQInput node is connected):
     a. Non-transactional messages are discarded.
     b. Transactional messages are rolled back to the input queue for retry.
        - If the backout count of the message is less than the backout threshold (BOTHRESH) of the input queue, the message is retried and re-sent to the out terminal.
        - When the backout count equals or exceeds the backout threshold, the message is placed on the backout queue (identified by the BOQNAME attribute of the input queue) if specified, or to the dead-letter queue if there is no backout queue defined or the MQPUT to the backout queue fails. If the MQPUT to the dead-letter queue fails, or there is no dead-letter queue defined, the message flow loops continuously trying to put the message to the dead-letter queue.
     c. If a path is defined for the failure (if both the catch terminal and the failure terminal are connected, the message is propagated through the catch terminal), that path defines the destination of the message.
6. If your message flow uses transaction mode=yes on the MQInput node properties, and the messages are not appearing on an output queue, check the path of the message flow.

   If the message flow has paths that are not failures, but that do not end in an output queue (or other persistent store), the message flow has not failed and the message is not backed out, or put to an alternative destination (for example, the catch terminal, the dead-letter queue, or the queue's backout queue).

   Check that all possible paths reach a final output node and do not reach a dead end. For example, check that:
   a. You have connected the unknown terminal of a Filter node to another node in the message flow.
   b. You have connected both the true and false terminals of a Filter node to another node in the message flow.

Your execution group is not reading messages from the input queues

- **Scenario:** Your execution group has started, but is not reading messages from the specified input queues.

- **Explanation:** A started execution group might not read messages from the input queues of the message flows because previous errors might have left the queue manager in an inconsistent state.
- **Solution:**
  1. Stop the broker, the WebSphere MQ listener, the WebSphere MQ channel initiator, and the WebSphere MQ queue manager.
  2. Restart the queue manager, the channel initiator, the listener, and the broker.

DataFlowEngine terminates while processing messages

- **Scenario:** While processing a series of messages, the DataFlowEngine process size grows steadily without levelling off. This might cause the DataFlowEngine process to terminate, if it cannot allocate any more memory, and restart. The message BIP2106E might be logged indicating the out of memory condition.

  In addition, if you are using DB2 on distributed platforms, you might get the message:

  ```
  SQL0954C  Not enough storage is  available in the application heap to process the statement.
  ```

  On z/OS, an SQLSTATE of HY014 might be returned with an SQL code of -99999, indicating that the DataFlowEngine process has reached the DB2 z/OS process limit of 254 prepared SQL statement handles.
- **Explanation:** When a database call is made from within a message flow node, the flow constructs the appropriate SQL, and this is sent using ODBC to the database manager. As part of this process, the SQL statement is prepared using the SQLPrepare function, and a statement handle is acquired so that the SQL statement can be executed.

  For performance reasons, after the statement is prepared, the statement and handle are saved in a cache to reduce the number of calls to the SQLPrepare function. If the statement is already in the cache, the statement handle is returned so that it can be re-executed with newly bound parameters.

  The statement string is used to perform the cache lookup. By using hardcoded SQL strings that differ slightly for each message, the statement is not found in the cache, and an SQLPrepare function is always performed (and a new ODBC cursor is opened). When using PASSTHRU statements, use parameter markers so that the same SQL prepared statement can be used for each message processed, with the parameters being bound at runtime. This approach is more efficient in terms of database resources and, for statements that are repeatedly executed, it is faster.

  However, it is not always possible to use parameter markers, or you might want to dynamically build the SQL statement strings at runtime. This potentially leads to many unique SQL statements being cached. The cache itself does not grow that large because these statements themselves are generally not big, but many small memory allocations can lead to memory fragmentation.
- **Solution:** If these types of situations are encountered, you can disable the caching of prepared statements by setting the MQSI_EMPTY_DB_CACHE environment variable to any value. When this environment variable has been created, the prepared statements for that message flow are emptied at the end of processing for each message. There might be a slight performance impact because every SQL statement is prepared.

Your execution group hangs or terminates with a core dump

- **Scenario:** While processing a message, an execution group either hangs with high CPU usage, or terminates with a core dump. The stack trace from the core dump or abend file is large, showing many calls on the stack. Messages written to the system log might indicate `out of memory` or `bad allocation` conditions.

The characteristics of the message flow in this scenario often include a hard-wired loop around some of the nodes.

- **Explanation:** When a message flow thread executes, it requires storage to perform the instructions that are defined by the logic of its connected nodes. This storage comes from the execution group's heap and stack storage. The execution of a message flow is constrained by the stack size, the default of which differs depending on the operating system.
- **Solution:** If a message flow of this magnitude is required, you can increase this limit and then restart the brokers running on the system to use the new value. For information on setting the stack size for your operating system, see System considerations for message flow development.

″Style sheet could not be processed″ errors are issued and your XMLTransformation node is not working after deployment

- **Scenario:** Your XMLTransformation node is not working after deploying resources and ″style sheet could not be processed″ errors are displayed.
- **Solution:**
  1. If the broker cannot find the style sheet or XML files required, migrate the style sheets or XML files with relative path references.
  2. If the contents of a style sheet or XML file are damaged and therefore no longer usable, for instance if a file system failure occurs during a deployment, redeploy the damaged style sheet or XML file.

Output messages are not sent to expected destinations

- **Scenario:** You have developed a message flow that creates a destination list in the LocalEnvironment tree. The list might contain queues for the MQOutput node, labels for a RouteToLabel node, or URLs for an HTTPRequest node. However, the messages are not reaching these destinations but there are no error messages.
- **Explanation:** There are several possible causes of this:
  – Check that you have set *Compute mode* to a value that includes the LocalEnvironment in the output message, for example `All`. The default setting of *Compute mode* is `Message`, and any changes that you make to LocalEnvironment are lost.

  – Check your ESQL statements. Because the content and structure of LocalEnvironment are not enforced, the ESQL editor (and Content Assist) does not provide any guidance for field references, and you might have incorrectly specified one or more of these references.

    Some example procedures to help you set up destination lists are provided in Populating Destination in the LocalEnvironment tree. You can use these procedures unchanged, or modify them for your own requirements.

You cannot determine which node is being referenced in your trace file

- **Scenario:** You have generated a service trace file for your message flow to trace the path of a message. However, you cannot determine which node is being referenced in the trace file.
- **Explanation:** In the trace file you might see text such as:

  ```
  Video_Test#FCMComposite_1_1 ComIbmMQInputNode , Video_Test.VIDEO_XML_IN
  ```

  In the line of trace text:

  Video_Test
  is the name of the message flow

FCMComposite_1_1
is the internal name for the node

ComIbmMQInputNode
is the type of node

VIDEO_XML_IN
is the node label, and is the name you see in your flow

The number on the end of the internal name is incremented; for example, FCMComposite_1_4 would be the fourth node you added to your flow. In the example, this section of the trace is referring to the first node in the message flow.

You cannot see any alerts when you change user trace

- **Scenario:** You cannot see any alerts for an execution group or message flow in the Alerts viewer when you change the user trace setting using the **mqsichangetrace**command.
- **Explanation:** No alert is generated when an execution group runs user trace at normal or debug level. Alerts are generated only for message flow trace. For message flow trace, the workbench is not notified of trace changes that are initiated by the **mqsichangetrace** command.
- **Solution:** Refresh the message flow, or disconnect and then reconnect to the domain, to see the alert.

Task list does not update when you make corrections to your files

- **Scenario:** The task list does not update any modifications that you make to an ESQL or Mapping file. You have made corrections to the files, and while there are no error flags in the file or file icon, the error remains as a task list item.
- **Solution:** To work around this problem, you need to set the following environment variable:

  ```
  JITC_COMPILEOPT=SKIP{org/eclipse/ui/views/tasklist/TaskListContentProvider}{resourceChanged}
  ```

You rename a flow that contains errors, but the task list entries remain

- **Scenario:** When you rename a message flow in the workbench for which there are error icons (red crosses) displayed on nodes and connections, those error icons are removed when changes are made. However, the task list entries remain.
- **Solution:** Refresh the Message Flow editor by closing and reopening it.

Terminals on a subflow get out of sync as changes are made

- **Scenario:** You have a message flow that contains subflow nodes. The name or number of terminals on the subflow gets out of sync as changes are made on the subflow itself. The same problem can happen with promoted properties.
- **Solution:** Refresh the Message Flow editor by closing and reopening it. Close the Message Flow editors containing subflows while the subflows are being changed.

The values of your promoted properties were lost after editing

- **Scenario:** You edited a message flow using the Message flow editor, and the values of your promoted properties were lost.

- **Explanation:** Due to a program defect, the values of promoted properties for nodes with more than a single subflow definition (that is, two identically named subflows in the same project reference path) are lost if the flow is edited and saved.
- **Solution:** To avoid this problem, ensure that each subflow in your project has a different name.

The Message Flow editor experiences problems when opening a Message Flow and opens in error mode

- **Scenario:** You attempt to open an existing Message Flow in the Message Flow Editor and it opens in read-only error mode displaying a list of parsing, or loading errors. The Message Flow is not opened and the following message is displayed:

  ```
  Invalid message flow file
  ```

- **Explanation:** The message flow file is unreadable or is corrupted, and the Message Flow editor cannot render the model graphically.
- **Solution:** Contact IBM Customer Support for assistance with the corrupted file.

You get an exception when saving changes to a .tblxmi file

- **Scenario:** You are using the RDB Table editor to make changes to a `.tblxmi`file, but keep getting exceptions when you try to save the changes.
- **Explanation:** When you make changes to a `.tblxmi`file using the Table editor, the editor saves the change successfully only for the first save. Subsequent saves after additional changes fail with an exception.
- **Solution:** To stop this from happening, close the Table editor after each save, and reopen it when you want to continue making changes.

You do not know when to use the MQeMbMsgObject object

- **Scenario:** You do not know when to use the MQeMbMsgObject object rather than the base MQeMsgObject object.
- **Explanation:** Any messages from existing WebSphere MQ Everyplace applications (which are of type MQeMsgObject, or a subclass) can pass through WebSphere Message Broker unchanged. WebSphere Message Broker cannot parse the contents of these messages. However, the full representation of the data that has been dumped is available within the broker. Do not operate on this data.
- **Solution:** If you need to parse or modify the data contained within a WebSphere MQ Everyplace message, use an MQeMbMsgObject object. This provides a parallel with standard WebSphere MQ messages: you can set fields such as correlation ID, and there is a field that can be parsed using any WebSphere Message Broker parser.

QoS1 and QoS2 SCADA messages are not being processed

- **Scenario:** The message flow that handles SCADA messages is not processing QoS1 and QoS2 messages.
- **Explanation:** The broker database, in which these messages are stored, is causing deadlock.
- **Solution:** If you are using DB2 as your broker database, turn off DB2 next key locking to avoid these deadlock problems. Issue the following command in a DB2 command window to make this change:

  ```
  db2set DB2_RR_TO_RS=YES
  ```

  You must now restart the DB2 database manager for this change to take effect.

Resolving problems when deploying message flows or message sets

1. To debug problems when deploying, check the logs:
   - The broker domain Event log
   - The local error log (the Windows Event log or the syslog)
   - The WebSphere MQ logs

   These logs might be on separate computers, and must be used in conjunction with the workbench output to ensure that the deployment was successful.

   Use the **mqsilist** command to check that the deployment was successful, or look in the Windows Event or broker domain Event log.
2. Use this checklist when you have deployment problems:
   - Make sure that the remote queue manager is running.
   - Make sure that channels are running.
   - Display the channel status to see if the number of system messages sent increases.
   - Check the channel from the remote end.
   - Check the queue manager name.
   - Determine whether the channel is a cluster channel.

This topic contains advice for dealing with some common problems that can arise when deploying message flows or message sets.

Preparing to deploy:

- "Migrated message flows are not available to add to a broker archive file" on page 49
- "An error is issued when you add a dictionary to a broker archive file" on page 49
- "You cannot drag and drop a broker archive file to a broker" on page 49
- "You cannot deploy a message flow that uses a plug-in message flow" on page 50
- "The compiled message flow (.cmf) file has not been generated" on page 50

During deployment:

- "The message flow deploys on the test system, but not elsewhere" on page 50
- "Your deployment indicates that the broker does not exist" on page 50
- "The Configuration Manager is trying to deploy to a broker that does not exist" on page 51
- "The Configuration Manager reports that it is out of memory" on page 51
- "A correlation name error is issued when you deploy to a Version 2.1 broker" on page 52
- "Error messages are issued when you deploy on z/OS" on page 52
- "Expected serialization of input is not occurring for a shared queue that serves multiple instances of a message flow on z/OS" on page 52
- "MQe nodes do not work as expected after deployment to a Version 6.0 broker" on page 53
- "Error messages are issued when you deploy" on page 53

- "Error message BIP2432E is issued when you deploy from a Version 6.0 Configuration Manager to a Version 2.1 broker" on page 57

Canceling deployment:

- "When canceling a deployment, **mqsilist** shows a message flow in an execution group but the tooling does not" on page 57

After deployment:

- "You are not notified of the result of a deploy" on page 57
- "You do not receive confirmation that the deployment was successful" on page 57
- "You cannot see any deployed message flows or message sets" on page 58
- "A deleted broker remains in the domains navigator" on page 58
- "Your XMLTransformation node does not work after deployment" on page 59

Migrated message flows are not available to add to a broker archive file

- **Scenario:** You have migrated your Version 2.1 and Version 5.0 message flows and message sets into the workbench using the **mqsimigratemsgflows** command and **mqsimigratemsgsets** command, but they are not available to add to a broker archive (bar) file.
- **Solution:** Migrated message flows and message sets are displayed in the workbench as a closed project. To make the message flows and message sets available to add to a bar file:
  1. Open the project by right-clicking the project and clicking **Open Project** from the **Project** menu.
  2. Clean the project by right-clicking the project and clicking **Clean**. This shows any errors, and makes the project resources available to add to a bar file.

An error is issued when you add a dictionary to a broker archive file

- **Scenario:** An error is issued when you add a dictionary to a broker archive (bar) file.
- **Explanation:** After you create a bar file and add a message set project to it, two files are created in the bar file: messageset.user.txt and messageset.service.txt. The user.txt file contains user log information, such as warning message BIP0177W, which states that the dictionary that you have created is not compatible with versions previous to WebSphere Message Broker Version 6.0. The information in this file should help you to diagnose the error. The service.txt file contains rudimentary dictionary information as used by the broker, and can be used by the IBM Support Center to diagnose problems.

You cannot drag and drop a broker archive file to a broker

- **Scenario:** You cannot drag and drop a broker archive (bar) file to a broker.
- **Explanation:** Bar files can be deployed only on an execution group. When you use drag and drop, you cannot drop the bar file onto a broker in the Broker Topology editor or in the Domains navigator.
- **Solution:** Drop the bar file onto an execution group in the Domains navigator, or select an execution group in the deploy dialog.

You cannot deploy a message flow that uses a plug-in message flow

- **Scenario:** You have created a message flow that contains an input node in a user-defined node project. However, you cannot deploy a message flow that uses this plug-in message flow.
- **Explanation:** Validation, compilation, and deployment do not recognize that a plug-in message flow contains an input node.
- **Solution:** To work around the problem, add a dummy Input node to the flow that you intend to deploy.

The compiled message flow (.cmf) file has not been generated

- **Scenario:** The compiled message flow file (.cmf) has not been generated. Therefore, it is not added to the broker archive file and cannot be deployed.
- **Explanation:** When you create mapping files, the overall file path length must not exceed 256 characters, due to a Windows file system limitation. If you try to add a message flow to a broker archive file with mapping or ESQL files with a path length that exceeds 256 characters, the compiled message flow is not generated and cannot be deployed.
- **Solution:** To overcome this problem, make sure that the following names are as short as possible:
  - The installation path
  - Any project names and broker schema names
  - Any ESQL and mapping file names

The message flow deploys on the test system, but not elsewhere

- **Scenario:** The message flow that you have developed deploys on the test system, but not elsewhere.
- **Solution:** Carry out the following checks:
  - Make sure that you have verified the installation by creating and starting a broker and deploying a single execution group. This shows that WebSphere Message Broker and the broker database are in place.
  - Ensure that the broker archive (bar) file's broker.xml file contains references to the correct resources for the new system.
  - Ensure that any referenced message sets are deployed.
  - If database resources or user-defined nodes are not accessible or authorized from the target system, the deploy fails. On distributed systems, ensure that your databases are defined as ODBC sources so that they can be accessed from WebSphere Message Broker. Also, set the broker's environment to allow access to the databases. (You might need to run a profile on Linux or UNIX systems.)
  - Any user-defined extensions that you are using in your message flow might not load if they cannot be found, or are not linked correctly. Consult the documentation for your operating system for details of tools that can help you to check your user-defined extension binaries.

Your deployment indicates that the broker does not exist

- **Scenario:** You have initiated deployment, but get an error indicating that the broker does not exist.
- **Explanation:** During the first stage of deployment, the Configuration Manager sends a configuration data stream to the SYSTEM.BROKER.ADMIN.QUEUE queue of each target broker. When the configuration data has been sent to all relevant brokers, control is returned to you. If an error is detected during this

first stage of deployment, the deployment is abandoned; no configuration data is sent to any broker and an error message is written to the log.

- **Solution:** Check that you have specified the correct queue manager associated with the broker. If it is correct, it is possible that the deployment message cannot be delivered because the queue to which it was sent (SYSTEM.BROKER.ADMIN.QUEUE) could not be found on the target queue manager. This returns a WebSphere MQ reason code of 2085.

  If that queue has been accidentally deleted, the broker environment has been damaged and you must repair it.

The Configuration Manager is trying to deploy to a broker that does not exist

- **Scenario:** The Configuration Manager is trying to deploy to a broker that does not exist.
- **Explanation:** This can happen for a number of reasons:
  - You deleted a broker component without removing from the Message Brokers Toolkit first.
  - You mistyped the name of the broker in the Message Brokers Toolkit, creating a reference to a broker that does not physically exist.
  - You did not follow the correct procedure for deleting a broker.
  - The broker can appear not to exist due to connectivity problems.
- **Solution:**
  1. Check that the broker does not exist by using the **mqsilist** command to list resources on the computer where the broker should exist.
  2. Check your configuration to ensure that the broker does not exist on a different computer in the domain.
  3. If you are confident that the broker does not exist and you want to remove the reference that is still in the Message Brokers Toolkit, follow the steps in Deleting a broker and Removing a broker from a broker domain.
  4. Remove completely all references to the broker from the Configuration Manager repository.

     You can do this by removing the broker from the topology using the Message Brokers Toolkit. Ensure that the broker's SYSTEM.BROKER.ADMIN.QUEUE queue has been deleted and initiate a further topology deploy.

     Alternatively, you can use the Configuration Manager Proxy API ″TopologyProxy.removeDeletedBroker(String)″ to programmatically purge all references to the broker:

     a. Start the Configuration Manager Proxy API Exerciser sample (which is available on the **Start** menu in Windows).

     b. Connect to the Configuration Manager.

     c. Right-click the topology object and click **Remove references to a previously deleted broker**.

The Configuration Manager reports that it is out of memory

- **Scenario:** The broker archive file is not deployed because the Configuration Manager claims to be out of memory, as indicated by error message BIP1111.
- **Explanation:** The most likely cause of this problem is that the Java Virtual Machine (JVM), which the Configuration Manager uses to perform the deployment logic, has run out of temporary storage space.

- **Solution:** Use the **maxJVMHeapSize** option on the Configuration Manager to increase the amount of memory allocated to the Configuration Manager's JVM:

  ```
  mqsistop ConfigMgr
  mqsichangeconfigmgr -j [newJVMHeapSizeMb]
  mqsistart ConfigMgr
  ```

  where *[newJVMHeapSizeMb]* is the size in megabytes to allocate; the minimum is 64.

  Retry the deploy operation. If the problem still occurs, increase the **maxJVMHeapSize** parameter further.

A correlation name error is issued when you deploy to a Version 2.1 broker

- **Scenario:** The following error message is issued when you deploy a broker archive file to a Version 2.1 broker:

  ```
  The correlation name 'CREATE' is not valid.
  Those in scope are: Environment, InputLocalEnvironment, OutputLocalEnvironment, InputRoot,
  InputBody, InputProperties, OutputRoot, InputExceptionList, OutputExceptionList,
  InputDestinationList, OutputDestinationList.
  ```

- **Explanation:** This error message indicates that you have attempted to deploy a message flow that contains a mapping node to a Version 2.1 broker. Mapping nodes are not supported on Version 2.1 brokers.
- **Solution:** The error can also be issued when you deploy a message flow that is in the same message flow project as another message flow that contains a mapping node to a Version 2.1 broker. If this is the case, move the message flow that contains a mapping node into a different project, and try the deployment again.

Error messages are issued when you deploy on z/OS

- **Scenario:** The following messages are written to the log when you deploy on z/OS:

  ```
  +(MQ05BRK) 0 BIP2070E: A problem was detected with WebSphere MQ while issuing
  MQPUT for WebSphere MQ queue SYSTEM.BROKER.ADMIN.REPLY, WebSphere MQ queue
  manager QM_01.  MQCC=2, MQRC=2030.
  +(MQ05BRK) 0 BIP2068E: The broker was unable to put an internal configuration
  message to message queue SYSTEM.BROKER.ADMIN.REPLY.
  ```

- **Explanation:** The transmission queue is not large enough for the messages issued by WebSphere Message Broker.
- **Solution:** Refer to the WebSphere MQ documentation for details on how to increase the size of the transmission queue.

Expected serialization of input is not occurring for a shared queue that serves multiple instances of a message flow on z/OS

- **Scenario:** Expected serialization of input is not occurring for a shared queue that serves multiple instances of a message flow on z/OS.
- **Explanation:** On z/OS, WebSphere MQ supports serialized access to shared resources, such as shared queues, through the use of a connection tag (serialization token) when an application connects to the queue manager that participates in a queue sharing group.

  There are a number of possible causes for this problem:

  – The user might have assigned different serialization tokens to the input nodes of the flows that get messages from the shared queue.

  – The message flows might be running in the same execution group. Serialization can only be effected between flows running in different

  execution groups, either on the same broker or on different brokers whose queue managers participate in the same queue sharing group.
  - The queue managers on which the brokers are running are not participating in a queue sharing group, or not participating in the same queue sharing group.
- **Solution:** Carry out the following checks:
  - Check that the same z/OS serialization token has been configured for the MQInput nodes in each flow using the shared queue.
  - Check that the message flows are running under different execution groups.
  - Check that the broker queue managers are part of the same queue sharing group and that there are no errors reported from the queue managers in the z/OS system log (SDSF log).

MQe nodes do not work as expected after deployment to a Version 6.0 broker

- **Scenario:** You have a message flow containing MQe nodes that works on a previous version broker, but when you deploy it to a Version 6.0 broker it fails.
- **Explanation:** The use of message flows that contain MQeInput and MQeOutput nodes in WebSphere Message Broker Version 6.0 is deprecated. It is still possible to use them to deploy from Version 6.0 to previous versions. When you deploy to a Version 6.0 broker, the MQe nodes are converted into MQ nodes.
- **Solution:** Redesign your message flow to use MQ nodes, and configure an external MQe gateway accordingly. As a route to migration you can deploy the message flow anyway, and create a suitable configuration to deal with this, as described in Configuring after MQe flow deployment. See full migration details in Migrating a flow containing WebSphere MQ Everyplace nodes.

Error messages are issued when you deploy

The error messages that might be generated during a deployment are listed below with an explanation:

Message BIP1106 with WebSphere MQ reason code 2030

- **Scenario:** Error message BIP1106 is issued with reason code 2030 when you are deploying a large message set.
- **Explanation:** The size of the message exceeds the maximum message length of the transmission queue to the broker queue manager.
- **Solution:** Increase the maximum message length for the transmission queue using the WebSphere MQ **alter qlocal** command, where the maximum message length is in bytes:

  ```
  alter ql(transmit_queue_name) maxmsgl(104857600)
  ```

  See the *WebSphere MQ System Administration Guide* for more information on this command.

Message BIP1106with WebSphere MQ error AMQ7463

- **Scenario:** Error message BIP1106 is issued with reason code 2102 when you are deploying a large message set to a broker that shares the queue manager with the Configuration Manager. WebSphere MQ error message AMQ7463 is also issued with the text: `The log for queue manager <`*`queue manager`*`> is full.`
- **Solution:**
  1. Stop the Configuration Manager using the **mqsistop** command.
  2. Stop the broker using the **mqsistop** command.

3. Stop the queue manager using the WebSphere MQ **amqmdain** command:
   ```
   amqmdain end <queue manager>
   ```
4. Start WebSphere MQ Services from the **Start** menu by clicking **Start** → **Programs** → **IBM WebSphere MQ** → **WebSphere MQ Services**.
5. Right-click the queue manager, and click **Properties**, then click the **Log** tab.
6. Increase the number of primary and secondary files so that the total size is larger than the deploy message.
7. Restart the Configuration Manager using the **mqsistart** command.
8. Restart the broker using the **mqsistart** command.
9. Restart the queue manager using the WebSphere MQ **amqmdain** command:
   ```
   amqmdain start <queue manager>
   ```

Message BIP1538E

- **Scenario:** Error message BIP1538E is issued with reason code 2218 when you are deploying a large message set.
- **Explanation:** The size of the message exceeds the maximum message size on the channel.
- **Solution:** Increase the channel **maxmsgl** parameter on both channel pairs, at both ends as follows:
  1. On the Configuration Manager queue manager, issue the WebSphere MQ **alter channel** command. For example:
     ```
     alter chl(CM_to_BRK) chltype(sdr) MAXMSGL(104857600)
     alter chl(BRK_to_CM) chltype(rcvr) MAXMSGL(104857600)
     ```
  2. On the Broker queue manager, issue the WebSphere MQ **alter channel** command. For example:
     ```
      alter chl(BRK_to_CM) chltype(sdr) MAXMSGL(104857600)
      alter chl(CM_to_BRK) chltype(rcvr) MAXMSGL(104857600)
     ```
  3. Stop and restart each of the channels for these changes to take effect.

Message BIP1536

- **Scenario:** You have defined a Configuration Manager to run with one user ID and you have defined a broker to run on a different computer with a different user ID. Deployment is successful but error message BIP1536 is issued when you deploy message flows and message sets to the broker.
- **Explanation:** The Configuration Manager is unable to register for internal subscriptions with the broker because the broker is running under one ID and the Configuration Manager is running under another ID. The broker and the Configuration Manager relay internal messages back and forth via publish/subscribe. These message are carried through WebSphere MQ, which requires certain authorizations.
- **Solution:**
  - Ensure that the broker's user ID is a member of the **mqm** and **mqbrkrs** groups.
  - Define the broker's user ID on the computer where the Configuration Manager is running.
  - Define the Configuration Manager's user ID on the computer where the broker is running.
  - Ensure that all IDs are in lowercase so that they are compatible between computers.

Message BIP1536 with message BIP7017

- **Scenario:** Error messages BIP1536 and BIP7017 are displayed.
- **Explanation:** The Configuration Manager has a problem registering its internal subscriptions on topics to do with broker status change, which it tries to do each time that you deploy a complete configuration. The cause of the problem is given by the message BIP7017, which indicates that you are running with a User Name Server configured, but that the broker to which you are deploying does not have the Configuration Manager service user ID in its user cache.
- **Solution:** Make the following checks. When you identify and correct the problem, the subscriptions will be registered correctly the next time you deploy:
  - Ensure that the User Name Server is started.
  - Ensure that the WebSphere MQ channels between the User Name Server and the broker are started.
  - Ensure that the user ID is present in the User Name Server's domain.

Message BIP1835

- **Scenario:** Error message BIP1835 is displayed.
- **Explanation:** The message set that you are deploying produces a message set dictionary larger than the internal limit of 4MB. This could be because you have many large message definitions defined to the same message set.

  The size of an exported message set is not a good indication of the size of the message set dictionary that is generated at the time of deployment, because the exported message set is stored as XML. This can be very verbose, but the dictionary has a much more compact internal format.
- **Solution:** Split the message definitions into several smaller message sets.

Message BIP2045

- **Scenario:** Error message BIP2045 is displayed.
- **Explanation:** This message indicates that an attempt has been made to assign a broker to a second (or re-initialized) Configuration Manager.

  Each broker is identified by a universally unique identifier (UUID), which is allocated when the Message Brokers Toolkit or Configuration Manager Proxy creates a definition for the broker. When the deployment occurs, a UUID check is made to help prevent accidental deployment of changes to brokers that are not under the control of the Configuration Manager. If you have received error BIP2045, the deployment message was addressed to a broker with one UUID, but the message was processed by a broker that was expecting a different UUID.
- **Solution:** If possible, re-issue the deployment request using the Configuration Manager that administers this broker.

  If this is not possible, use the Configuration Manager Proxy API BrokerProxy.setUUID() method to set the UUID of the Configuration Manager's broker definition to be the UUID that the broker is expecting. This API can also be invoked using the Configuration Manager Proxy API Exerciser sample.

  Invoking this method allows the Configuration Manager to deploy to the broker, although any resources that were previously deployed to the broker are not monitored by the Configuration Manager (unless they were already being monitored).

  If you are still unable to deploy, follow these steps:

  1. Delete the broker component, its configuration tables, and its queue manager.
  2. Remove the broker definition from the domain using the Message Brokers Toolkit or Configuration Manager Proxy.
  3. Redeploy the topology.

4. Re-create the broker component and add a reference to it using the Message Brokers Toolkit or Configuration Manager Proxy.
5. Redeploy any resources to the broker.

To prevent this problem from recurring when you delete or re-create brokers, always remove the broker definition from the domain and deploy the topology successfully before the running broker component is stopped and deleted.

Message BIP2066

- **Scenario:** Error message BIP2066 is displayed.
- **Explanation:** The deployment request was not acknowledged by the execution group before the broker timeout **ConfigurationTimeout** plus the **ConfigurationDelayTimeout** (default 60 seconds) expired.
- **Solution:** Change these timeouts using the **-g** and **-k** flags of the **mqsicreatebroker** and **mqsichangebroker** commands.

Message BIP2242

- **Scenario:** You see error message BIP2242.
- **Explanation:** The deploy (configuration change) request was not accepted before the broker timeout *ConfigurationTimeout* (default 300 seconds) expired. The timeout needs to be long enough for the message flow to complete processing its current message and then accept the deploy request.
- **Solution:** See Commands for details of how to set these timeouts using the**-g** and **-k** flags of the **mqsicreatebroker** and **mqsichangebroker** commands.

Tagged/Delimited String (TDS) Validator error

- **Scenario:** You try to deploy a message set with a TDS wire format that has an error.
- **Explanation:** The following extract from an error log illustrates what you might see for a TDS Validator error. In this case, the cause of the problem is that the element `Town` does not have a tag defined.

```
TDS Extractor Trace File
========================

Beginning Extract..

Extracting Identification Info
Extracting Project Info
Extracting Messages
Extracting Elements
Extracting Compound Types
Extracting Type Members
Extracting Type Members
Extracting Type Members
Extracting Type Members
Extracting Type Members
Beginning Indexing..

Creating Member IDs to Tags Index Table.

Beginning Validation..

Validating Project
Validating Types
ERROR: TDSValidator::ValidateTypeMemberSimpleElement:
  Simple elements in a type with Data Element Separation attribute = Tagged
  Delimited must have the following attribute set:
  Element Level - Tag
(Element ID: Town)
```

```
(Type ID: AddressType)
Return Code: -80

Validating Messages

Trace Info
===========
EXCEPTION: TDSValidator::Validate:
  TDS Validation failed.
    1 errors
    0 warnings
Return Code: -1
```

- **Solution:** Use the information in the error log to rectify the problem.

Error message BIP2432E is issued when you deploy from a Version 6.0 Configuration Manager to a Version 2.1 broker

- **Scenario:** Error message BIP2432E is issued when you deploy from a Version 6.0 Configuration Manager to a Version 2.1 broker.
- **Explanation:** This error is caused by the Version 6.0 Configuration Manager sending Version 6.0 ESQL to the Version 2.1 broker.
- **Solution:** When you add ESQL to a bar file that is being deployed to Version 2.1, select the **Compile ESQL for broker Version 2.1** check box in the Broker Archive editor. You need to have different bar files for deployment to Version 6.0 and Version 2.1.

When canceling a deployment, mqsilist shows a message flow in an execution group but the tooling does not

- **Scenario:** When canceling a deployment, **mqsilist** shows a message flow in an execution group but the tooling does not show it. You have canceled deployment to an execution group, but the deployment takes place and is shown by **mqsilist**.
- **Solution:** To fix this problem, delete the execution group, but be aware that **mqsilist** still shows the old execution group if it is the only one that has been deployed. If you re-create the execution group in the tooling, it is listed twice by **mqsilist** until the broker is recycled.

You are not notified of the result of a deploy

- **Scenario:** After you have closed the Broker Topology editor, saved your changes, and requested a deploy, you are not notified of the result of a configuration deploy.
- **Explanation:** When you close the editor, the notification link with the Configuration Manager is broken, so the workbench is not notified of the deploy result. This is the same for any eventual failure notifications.
- **Solution:** To ensure that you receive any notification, save the editor without closing it and wait for the alert `Waiting for answer from Configuration Manager` to disappear. This alert is removed automatically when the editor receives all expected responses from the Configuration Manager.

You do not receive confirmation that the deployment was successful

- **Scenario:** You have initiated deployment but have not received confirmation that the deployment was successful.
- **Solution:** Assume that communications from the workbench to the Configuration Manager are working, because the workbench gives an error immediately if this link fails. Follow these steps to find out if there is a communication problem that stops the receipt of deployment confirmation:

1. Stop the broker using the **mqsistop** command.

   On the broker queue manager, check the SYSTEM.BROKER.ADMIN.QUEUE queue. If there are messages on this queue, there might be a broker error; look in the local error log on the broker's system. If the queue is empty, continue with the next step.
2. Redeploy your message flow.

   Recheck the SYSTEM.BROKER.ADMIN.QUEUE queue on the broker queue manager. If the queue depth has not increased, there is a problem with the channel between the Configuration Manager and the broker; check the WebSphere MQ logs. If the queue depth has increased to one, continue with the next step.
3. Stop the Configuration Manager, then start the broker.

   This allows the broker to process the configuration change and send a response. Wait for the broker to process the message; this could take some time. (If the message takes more than five minutes to process, there is a problem with the deployed message and failure messages appear in the local error log on the broker's system indicating the reason for the failure.) Check the SYSTEM.BROKER.ADMIN.REPLY queue on the Configuration Manager's queue manager. This should contain a response. If it does not, check that the channel from the broker to the Configuration Manager is running.
4. Restart the Configuration Manager.

   Check that the messages are read from the SYSTEM.BROKER.ADMIN.REPLY queue on the Configuration Manager's queue manager. If they are not, check the local error log on the Configuration Manager's system for errors.
5. Refresh the Event Log editor in the workbench. You should now get a reply.

You cannot see any deployed message flows or message sets

- **Scenario:** After you deploy a broker archive (bar) file, you cannot see any deployed message flows or message sets under the execution group.
- **Explanation:** After deployment, you should get an information dialog that tells you whether deployment has been started successfully on the Configuration Manager. When the deploy has started, the time it takes to complete depends on your configuration size and work overload on your computers. However, if after a while no deployment-related messages are displayed in the Event Log editor, this means that the broker has not responded to the deployment request.
- **Solution:** Check that the broker is running and that all WebSphere MQ queue managers and channels between the Configuration Manager and the broker are running. If the messages are present in the Event Log editor, see if any errors have occurred that mean that your changes were not activated.

A deleted broker remains in the domains navigator

- **Scenario:** After you have deleted a broker and deployed changes, the broker remains in the domains navigator.
- **Explanation:** If you did not receive any failure response indicating that the deletion failed, and the Event Log editor has been automatically refreshed with a set of messages about the deployment result, the deletion succeeded but the workbench has not been notified of this deletion.
- **Solution:** For the broker topology to reflect the actual Configuration Manager content, refresh the broker topology. If the broker topology is open, click **Revert**.

Your XMLTransformation node does not work after deployment

There are two scenarios that explain why your XMLTransformation node might not work after deployment:

Error messages are displayed indicating that the style sheets were not found

- **Scenario:** Error messages are displayed indicating that the style sheets were not found.
- **Explanation:** This error message is displayed if the broker cannot find the style sheets or XML files required, or if the content of a style sheet or XML file is damaged and therefore no longer usable. This can happen if a file system failure occurs during a deployment.
- **Solution:** If the style sheets or relevant XML files cannot be found, perform the migration steps described in Migration of style sheets and XML files.

  If the style sheets or XML files are damaged, redeploy the damaged style sheets or XML files.

You get unexpected transformation results.

- **Scenario:** You get unexpected transformation results.
- **Explanation:** For complex message flows, incompatibility might arise among style sheets and XML files after a deployment. There are two likely reasons for this:
  - Only part of the cooperating style sheets or XML files are deployed and updated (this might be caused by a file system failure).
  - Multiple XMLT nodes that are running inside the same execution group are supposed to use compatible style sheets, but are using different versions to process the same incoming message.
- **Solution:** If only part of the cooperating style sheets or XML files are deployed and updated, resolve any incompatibility by redeploying the compatible versions. To avoid multiple XMLT nodes using different versions of the style sheet, pause relevant message flows in the target execution group before performing the deployment, then restart the flows.

Resolving problems when debugging message flows

This topic contains advice for dealing with some of the common problems that can arise when debugging message flows.

Starting and stopping the debugger:

- “Java class error messages are displayed when you start the debugger” on page 60
- “An endless ″waiting for communication″ message is displayed when you start the debugger” on page 60
- “The debugger seems to stop” on page 61
- “The session ends abnormally while debugging” on page 61
- “An error message is displayed indicating that the IBM Agent Controller is not installed” on page 61
- “Message flow engines are not available for selection” on page 61
- “You cannot see the list of execution groups” on page 61
- “You see the wrong execution group names in the agent page” on page 61
- “Shared Memory Allocation Error on AIX” on page 62

- “An error message is displayed indicating that the debug session cannot be launched” on page 62

During debugging:

- “A timeout occurs while waiting for the IBM Rational Agent Controller service to connect” on page 62
- “The debugger does not pause at the next breakpoint” on page 63
- “The message does not stop executing at any breakpoint” on page 63
- “Editing problems occur in the Message Flow editor” on page 63
- “Editing the MQ message descriptor (MQMD) causes unexpected behavior in the debugger” on page 63
- “You cannot see the message content when debugging your message flow” on page 63
- “You cannot see the message flow names in the Debug view” on page 63
- “You cannot see the deployed flow names in the Debug view” on page 64
- “There is an exclamation mark above a node during debugging” on page 64

After debugging:

- “You cannot change a message flow after debugging” on page 64
- “You redeployed a debugged message flow but deployment hangs” on page 64

Java class error messages are displayed when you start the debugger

- **Scenario:** You are trying to start the debugger on a message flow, but the debugger does not start, and a number of event errors are issued about Java classes.
- **Explanation:** The most likely cause of this problem is that you have not installed the IBM Agent Controller. Although the Agent Controller is not a prerequisite for WebSphere Message Broker, it is a prerequisite for the message flow debugger.
- **Solution:** Install the IBM Agent Controller.

An endless ″waiting for communication″ message is displayed when you start the debugger

- **Scenario:** After you click **Start Debugging**, you get an endlessly cycling progress bar entitled ″waiting for communication″. You never get the message ″debug session started″ in the information pane.
- **Explanation:** If the flow has nodes with ESQL statements, the flow might not deploy even if the statements are syntactically correct. This can occur, for example, because of multiple declarations or uninitialized variables (that is, semantic problems that the syntax parser does not pick up). Always check the workbench Event log to confirm that the debugged version of your flow has deployed; it has the same name as the original flow, with the suffix `_debug_` added.

  If the flow does not deploy properly, the debugger cannot establish communication with the flow, and you get the endless progress bar.
- **Solution:** Click **Cancel** to clean up and return to a good state, then fix your errors and try again. As a check, see if your flow can deploy without the debugger.

The debugger seems to stop

- **Scenario:** You are debugging a flow and continue after encountering a breakpoint. However, nothing seems to happen and after about a minute, a progress bar appears, indicating that the debugger is waiting for communication.
- **Explanation:** There are two possibilities.
  - The message flow might have encountered a time-intensive operation, such as a huge database query that you simply have to wait for.
  - The broker ended, or some other extraordinary condition occurred, and communication was lost. In this case, click **Cancel** to stop the debug session.

The session ends abnormally while debugging

- **Scenario:** After debugging a message flow, the session ends abnormally and you still have the debug instance of the message flow (mf_debug_) deployed to the broker's execution group. You are concerned that this is going to affect the operation of the flow, and want to put the execution group back to its original state.
- **Explanation:** The orphaned message flow should behave as the flow would have done normally, and the Debug nodes have no effect on message processing. If you have a small number of nodes in the flow, corrective action makes no noticeable difference to the flow, apart from its name. However, if you have a large flow (that is, more than 15 nodes or several subflows), take the corrective action described below, because the performance of message processing might be affected.
- **Solution:** A full redeploy of the broker should replace the orphaned flow with the original message flow. If this has no effect, remove the orphaned flow from the execution group, deploy, and then add the flow and deploy to restore the original state of the broker as it was before the debugging session.

An error message is displayed indicating that the IBM Agent Controller is not installed

- **Scenario:** You are using the message flow debugger and an error is issued indicating that the IBM Agent Controller is not installed, or that you have chosen the wrong hostname or port. You have the IBM Agent Controller service started, and the hostname and port are valid.
- **Solution:** Close and reopen the workbench, and retry the command. You can also try stopping and restarting the IBM Agent Controller service.

Message flow engines are not available for selection

- **Scenario:** You open the **Attach to the Message Flow Engine** wizard, but there are no message flow engines listed for the host computer.
- **Solution:** Close the wizard, restart IBM Agent Controller on the server computer, then open the wizard again. See also Attaching to the flow engine.

You cannot see the list of execution groups

- **Scenario:** Your Agent Controller is started, your broker is running, but you cannot see the list of execution groups in the Agent page when you attach to the debugger.
- **Solution:** Start your Agent Controller services before you start the broker. Restart the Agent Controller and try to attach again.

You see the wrong execution group names in the agent page

- **Scenario:** You see the same execution group names in the Agent page when you try to attach to the debugger.

- **Explanation:** The Agent Controller has not updated the agent list since the last attempt to attach to the debugger.
- **Solution:** Restart the Agent Controller to refresh the list.

Shared Memory Allocation Error on AIX

- **Scenario:** Your Agent Controller has started, your broker is running, and you get an error message saying that shared memory allocation has failed after the broker is attached to the Agent Controller.
- **Explanation:** This is a general timing problem that occurs when the Agent Controller is connected to the broker when the broker has not started completely.
- **Solution:** Wait until the broker has started completely before attaching it to the Flow Debugger. Alternatively, set the logging level in the Agent Controller to debug or information; this allows more time for the broker to start up. The following steps show you how to change the logging level.
  1. Go to the IBM Agent Controller install dir/config directory and open the configuration file serviceconfig.xml.
  2. Change the **loggingLevel** tag to debug or information. The default value is warning.
     ```
     <?xml version="1.0" encoding="UTF-8"?>
     <!DOCTYPE AgentControllerConfiguration SYSTEM
     "../dtd/serviceconfig.dtd">
     <AgentControllerConfiguration
     activeConfiguration="default" loggingDetail="LOW"
     loggingLevel="DEBUG" port="10002" secureport="10003"
     version="5.0.0">
     ```
  3. Restart the Agent Controller.

An error message is displayed indicating that the debug session cannot be launched

- **Scenario:** You try to relaunch or invoke a new debug session but when you click the green **Debug** icon🐞, an error message is displayed stating: `Cannot launch this debug session`.
- **Explanation:** When you click on the **Debug** icon, it re-launches the last debug session. It fails if you have not created a debug session previously. It also fails if the broker and execution group previously attached to in a debug session are no longer running or have been restarted; the session cannot be re-attached without re-selection of the new broker and execution group process instance.
- **Solution:** Close the error message and click the drop-down arrow immediately to the right of the **Debug** icon. Re-select or modify the broker and execution group information in the previous debug launch configuration by clicking **Debug** in the drop-down menu and selecting the previous debug launch configuration. See Attaching to the flow engine for more information.

A timeout occurs while waiting for the IBM Rational Agent Controller service to connect

- **Scenario:** You see error messages indicating that the IBM Rational Agent Controller service failed to start and that a timeout occurred while waiting for the Agent Controller service to connect.
- **Explanation:** The IBM Rational Agent Controller might be using the wrong version of the JVM.
- **Solution:** Ensure that a supported JVM is being used. To determine which JVM is being used, issue the java -version command at the command line. To get the

correct result, this command must call the Java executable file that was specified for use when the Agent Controller was installed.

The debugger does not pause at the next breakpoint

- **Scenario:** The message flow debugger does not pause at the next breakpoint in your message flow.
- **Solution:** Perform the following checks:
  1. Check whether your DataFlowEngine is running, and if it is not, restart it.
  2. Check the input queue. If your input queue has the leftover messages from the previous time that you used the debugger, you need to clear them before you send a new message.

The message does not stop executing at any breakpoint

- **Scenario:** The message does not stop executing at any breakpoint after you attach to the debugger.
- **Explanation:** This could be a timing problem.
- **Solution:** Restart the broker, wait until the broker has started completely and try to attach the debugger again.

Editing problems occur in the Message Flow editor

- **Scenario:** Editing problems occur when you are using the Message Flow editor while debugging a message flow.
- **Solution:** Do not attempt to edit the message while the flow debugger is attached. If you need to edit a message flow, detach the debugger, edit the message flow, and then redeploy the message flow.

Editing the MQ message descriptor (MQMD) causes unexpected behavior in the debugger

- **Scenario:** You edit properties of the message MQMD descriptor in the Message Set editor, but this causes unexpected behavior in the debugger.
- **Explanation:** If you edit the content of the MQMD descriptor, these fields take a certain range of values. You need to know these ranges before editing the properties. Unless you explicitly specify the value of these fields, they take default values, and certain fields might not have been specified in the message. The values in the fields that are not explicitly set in the message are default values; do not change these unless you are aware of their importance or the possible range of values.

You cannot see the message content when debugging your message flow

- **Scenario:** You are using the message flow debugger, and you can see the message passing through the message flow, but you cannot see the content of the message.
- **Solution:** Open the Flow Debug Message view by clicking **Window** → **Show View** → **Other** → **Message Flow** → **Flow Debug Message**, then click **OK**.

You cannot see the message flow names in the Debug view

- **Scenario:** You cannot see the deployed message flow names in the Debug view after attaching the debugger to the execution group.
- **Solution:**
  1. Stop the broker on which the execution group is running.
  2. Restart the IBM Agent Controller that is running on the same computer as the broker.

3. Restart the broker.

You cannot see the deployed flow names in the Debug view

- **Scenario:** You cannot see the deployed flow names in the Debug view after attaching to the execution group.
- **Explanation:** This could be a timing problem.
- **Solution:** Wait until the broker has started completely and try to attach the debugger again, or restart the Agent Controller that is running on the same computer as the broker, then restart the broker.

There is an exclamation mark above a node during debugging

- **Scenario:** In the Message Flow editor, an exclamation mark (!) is displayed above a node during debugging.
- **Explanation:** An exception has occurred in the node during debugging.
- **Solution:** Look under the ExceptionList in the Variables view of the Debug perspective to find out what error has occurred.

You cannot change a message flow after debugging

- **Scenario:** You were debugging, but now your message flow seems to be stuck. When you put a new message in, nothing happens.
- **Explanation:** This might be because a message was backed out, but you have not set the **Backout requeue name** property of your input queue.
- **Solution:** Set the **Backout requeue name** property to a valid queue name (such as the name of the input queue itself) and your flow will become unstuck.

You redeployed a debugged message flow but deployment hangs

- **Scenario:** You found problems in your message flow by using the debugger. You changed your message flow and then redeployed it, but now the deployment hangs.
- **Solution:** Make sure that when you redeploy the flow to an execution group, the execution group is not still attached to the debugger.

Resolving problems when developing message models

This topic contains advice for dealing with some common problems that can arise when working with message sets.

Message definition files:

- "Your message definition file will not open" on page 65
- "A message definition file error is written to the task list" on page 65
- "Your physical format property values have reverted to defaults" on page 65
- "You are unable to enter text in the Message Definition editors" on page 66
- "Objects in your message definition file are not listed in alphabetical order" on page 66
- "You want to make the Properties tab the default tab in the message definition editor" on page 67
- "Error messages are written to the task list after you have imported related XML schema files" on page 66

- "A warning is written to the task list because a group contains two different elements with the same XML name in the same namespace" on page 66
- "You are unable to set up a message definition file to include a message definition file within another message definition file" on page 67
- "Error message BIP5410 is issued because a union type cannot be resolved" on page 67
- "Error message BIP5395 is issued because an xsi:type attribute value does not correspond to a valid member type of the union" on page 68
- "Error message BIP5396 is issued because a data type does not correspond to any of the valid data types of the union" on page 68
- "A warning is issued because a union type contains two or more simple types derived from the same fundamental type" on page 69
- "An error message is issued because a list type is based on a union that also contains a list type" on page 69
- "An error message is issued because a union type contains an enumeration or pattern facet" on page 69

Other:

- "A deprecation error is issued on an imported .mrp file" on page 69
- "Your imported message set looks like a simple project" on page 70
- "User trace detects an element length error" on page 70

Your message definition file will not open

- **Scenario:** You have created a complex type with a base type that causes the types to be recursive, and now your message definition file will not open.
- **Solution:** Recover the last saved version of your message definition file from the local history or from your repository.

A message definition file error is written to the task list

- **Scenario:** An error is written to the task list indicating that the message definition file has been corrupted.
- **Explanation:** This error message appears when the base type of the complex type is itself, or a circular definition of two or more simple types. This type of recursion is not supported.
- **Solution:** Examine your code and ensure that the type of recursion described above does not occur.

Your physical format property values have reverted to defaults

- **Scenario:** Someone has sent you a message definition file. You have added it to a message set project, but all the physical format property values have reverted to defaults.
- **Explanation:** You cannot transfer message definition files on their own in this way because these files are not entirely independent. It is the messageSet.mset file that lists all the physical formats that are applicable to a message set. For example, if you have a message set *A* with CWF format *CWF1*, and a message set *B* without any physical formats, a message definition file copied from *A* to *B* does not show *CWF1* as a known physical format (although the properties are still in the message definition file).
- **Solution:** If possible, request that you are sent the entire message set project.

  If this is not possible, add physical formats to the receiving message set so that they match the originating message set. Then any dormant properties become visible. However, make sure that the message set level physical format properties match, or those object properties that inherit defaults from the message set level will be incorrect.

You are unable to enter text in the Message Definition editors

- **Scenario:** You are unable to enter text in the Message Definition Overview editor, or change the text of an enumeration or pattern facet in the Message Definition Properties editor.
- **Explanation:** The Message Definition editors are cell editors, and to enable them you have to double-click the cell. The first click selects the cell; the second click puts the cell into active state, allowing you to edit it.

Objects in your message definition file are not listed in alphabetical order

- **Scenario:** The objects in your message definition file (.mxsd) are listed in the order that you created them, but you want them to be in alphabetical order.
- **Solution:** In the Broker Application Development perspective, double-click the message definition file to open it in the Outline view, then click **az** (at the top of the Outline view). This changes the order of objects within each of the message definition file's folders (the Messages, Types, Groups, Elements, and Attributes folders) to be alphabetical.

Error messages are written to the task list after you have imported related XML schema files

- **Scenario:** When you have finished importing a collection of related XML schema files, you find that the workbench task list contains error messages for the message set project, indicating that referenced types or other objects cannot be found.
- **Explanation:** This is typically an indication that one message definition file includes or imports another message definition file, but that the Message Definition editor is unable to resolve the specified link.
- **Solution:**
  - Check that a message definition file exists in the workbench for each of the related XML schema files that you are importing. If the new message definition files have been created, they will appear in the Resource Navigator.
  - Using the Properties Hierarchy in the Message Definition editor, check that any message definition files that include or import other message definition files are correctly locating the target files.

  One common scenario is where you have two XML schema files *X* and *Y*, which both exist in the same directory in the file system but where *X*, which includes *Y*, is defined with a real target namespace, whilst *Y* is defined in the notarget namespace. After you import the two files, *X* is placed into the location determined by its namespace, but *Y* is placed into the default namespace location used for files defined in the notarget namespace. This causes the link between the two files to break, and you must modify *X* so that it correctly includes *Y* in its new relative location.

A warning is written to the task list because a group contains two different elements with the same XML name in the same namespace

- **Scenario:** A warning is written to the task list because a group contains two different elements that have the same XML element name, in the same namespace.

- **Explanation:** When a message that has an XML physical format is validated, the validation includes a test to identify any part of a message definition where the parser might not be able to uniquely determine which element is represented by the XML name. This test generates a warning when a group contains two different XML elements with the same element name in the same namespace, even in cases where the duplication is legitimate.
- **Solution:** Determine whether the duplication identified in the warning message is in fact a problem that needs to be corrected, or whether it has arisen because of two elements on opposite sides of a choice sharing the same XML name, in which case the duplication is legitimate because no ambiguity can occur.

You are unable to set up a message definition file to include a message definition file within another message definition file

- **Scenario:** You are unable to set up the include property of a message definition file to include a second message definition file that is included within another message definition file.
- **Explanation:** A message definition file can only reference objects in another message definition file if the first file references the second file directly. For example, if there are three message definition files, *A*, *B* and *C* where *A* includes *B* and *B* includes *C*, then *A* can reference objects in *B* and *B* can reference objects in *C*, but *A* cannot reference objects in *C*.

  You might also encounter this problem after importing XML schema files that have nested includes.
- **Solution:** You can work around this problem by including the message definition file directly, which, in the above example, would mean including *C* directly in *A*. Alternatively, you can define all the message set definitions that are in *C* directly in *B* and then delete *C* so that *A* needs to include only *B*.

You want to make the Properties tab the default tab in the message definition editor

- **Scenario**: When using the Message Definition editor to edit your message definition files, you prefer to use the **Properties** tab rather than the **Overview** tab, but the **Overview** tab is always selected as the default.
- **Solution:** Use the workbench preferences to choose the **Properties** tab as the default tab:
  1. From the menu click **Windows** → **Preferences**.
  2. Expand **Message Set**.
  3. Select **Editors**.
  4. In the **Message Definition tab order** section, select the **Properties then Overview** radio button.
  5. Click **OK**.

Error message BIP5410 is issued because a union type cannot be resolved

- **Scenario:** Error message BIP5410 is raised indicating that an element or attribute is based on a union type and that the element or attribute could not be cast to any member of the union.
- **Explanation:** When parsing an element or attribute that is based on a union type, the MRM XML parser uses an xsi:type attribute, where present, to resolve the union.

  If an xsi:type attribute is not present, or an attribute is being parsed, the parser tries each union member type in turn, attempting to cast to the associated simple type, until the cast succeeds. The order of precedence for the attempted cast is the order in which the member types are listed in the message model, under the union type, in the Outline view.

If the data cannot be cast to any of the simple types within the union, the union cannot be resolved and a parser error is reported.

- **Solution:** Perform the following checks:
  - Check that the message contains a valid value for the element or attribute identified in the error message.
  - Check that the member types of the union identified in the error message contain the correct list of simple types.
  - If possible, use an xsi:type attribute to resolve the union explicitly.

Error message BIP5395 is issued because an xsi:type attribute value does not correspond to a valid member type of the union

- **Scenario:** Error message BIP5395 is raised indicating that an element is based on a union type that has an xsi:type attribute with a value that should resolve the union explicitly to one of its modelled member types and that the xsi:type attribute value does not correspond to a valid member type of the union.
- **Explanation:** When parsing or writing an element or attribute that is based on a union type, the MRM XML parser or writer uses an xsi:type attribute, where present, to resolve the union. The parser resolves the value of the xsi:type attribute to a simple type in the dictionary and checks that the simple type is a valid member type of the union.

  If the xsi:type attribute identifies a type that is not a member type of the union, it reports an error.
- **Solution:** Perform one of the following steps:
  - Modify the message so that the xsi:type attribute identifies a valid member type of the union.
  - Check that the member types of the union identified in the error message contain the correct list of simple types.

Error message BIP5396 is issued because a data type does not correspond to any of the valid data types of the union

- **Scenario:** Error message BIP5396 is issued indicating that an element or attribute is based on a union type but the data type does not correspond to any of the valid data types in the union.
- **Explanation:** When writing an element or attribute that is based on a union type, the MRM XML writer uses an xsi:type attribute, where present, to resolve the union. If an xsi:type attribute is not present, the writer tries to match the data type of the literal value in the tree to a simple type in the union. If the literal value cannot be matched to any of the simple types in the union, it reports a writing error.
- **Solution:** Perform one of the following steps:
  - Check that the message or ESQL contains a valid value for the element or attribute.
  - Check that the union type on which the element is based contains the correct list of simple types.
  - Consider using an xsi:type attribute to resolve the union explicitly.
  - Consider changing the type of the element in the tree to correspond with one of the union member types.

A warning is issued because a union type contains two or more simple types derived from the same fundamental type

- **Scenario:** A warning is issued during local validation indicating that a union type contains two or more simple types derived from the same fundamental type.
- **Explanation:** The broker does not apply value constraints until the data is in the logical tree. Therefore, it is not possible to choose between two simple types that are derived from the same fundamental type but with different constraints. For example, the broker will not be able to differentiate between a member type of integer with a range of 1-10 and another member type of integer with a range of 11-20 so will resolve the union to the first member type of integer.
- **Solution:** Ensure that the union type identified in the warning does not contain more than one simple type derived from the same fundamental type or ensure that the ordering of such memberTypes is as desired.

An error message is issued because a list type is based on a union that also contains a list type

- **Scenario:** An error message is issued during logical validation indicating that a list type is based on a union type that includes a list type as one of its member types.
- **Explanation:** Lists of lists are not legal. An item type of a list type cannot be a list type itself nor can it be derived at any level from another list type. Therefore, a list type cannot have an item type of a union type that includes a list type as one of its member types.
- **Solution:** Ensure that any union type specified as an item type for a list type does not include a list type as one of its member types.

An error message is issued because a union type contains an enumeration or pattern facet

- **Scenario:** An error message is raised during logical validation indicating that a union type contains an enumeration or pattern facet that is not supported because enumeration and pattern facets must be specified directly on the member type.
- **Explanation:** The broker cannot support the union type facets of pattern and enumeration applied directly to a restriction of a union type. It can only support facets directly on the chosen member type. The Message editor provides support for these facets to enable the import of schemas using direct facets on a restriction of a union type but it issues a warning notifying that they will be ignored by the broker.
- **Solution:** If you want to use the enumeration or pattern facets, specify them directly on the member type and not on the union type itself.

A deprecation error is issued on an imported .mrp file

- **Scenario:** You have imported an `.mrp` file, and get an error message indicating that complex elements or embedded simple types are deprecated.
- **Explanation:** Complex elements and embedded simple types have no exact equivalent in XML schema. The closest equivalent to a complex type in XML schema is to derive a complex type from a simple type. However, such a type is not allowed to contain elements; only attributes are allowed.
- **Solution:** If your complex type contains elements that cannot be modelled as attributes, set the **Mixed content** flag on the complex type. If you need to parse messages that contain mixed content, set the **Mixed content** flag on the parent

complex type. The anonymous data that used to be modelled by the complex type or the embedded simple type then appears as a self-defining node in the parsed message tree.

Your imported message set looks like a simple project

- **Scenario:** You have imported a Version 2.1 message set into the workbench using the **mqsimigratemsgsets** command, but, when you open the project, the content organization looks more like a simple project.
- **Explanation**: Occasionally, a message set project is displayed like a simple project. If you look at the project icon, it does not contain an envelope.
- **Solution:** Close the project and reopen it.

User trace detects an element length error

- **Scenario:** You have defined an MRM message and when you execute the message flow, an error message is written to the user trace indicating that the length of an element is invalid.
- **Explanation:** You have added an element of type string to the complex type making up the message, but you have not defined the length of the string in the instance of the element in its type.
- **Solution:**
  1. Open the message definition file in the Message Definition editor.
  2. In the Outline view, expand the appropriate complex type, then click the element to display its properties in the editor area.
  3. In the Properties Hierarchy, make sure that the element is selected within the CWF physical format under **Physical Properties**.
  4. Specify the properties that this element will have when it is part of the complex type; for example, in the above case, make sure that the **Length Count** value is set.
  5. Save any changes in the Message Definition editor and redeploy the message.

Resolving problems when using messages

This topic contains advice for dealing with some common problems that can arise when using messages:

- "A communication error is issued when you use the enqueue facility" on page 71
- "The enqueue facility is not picking up changes made to a message" on page 71
- "You do not know which header elements have any effect in enqueue" on page 71
- "Enqueue message files are still listed after they have been deleted" on page 72
- "The ESQL transform of an XML message gives unexpected results" on page 72
- "Error message BIP5005 is issued by the Compute node" on page 72
- "A message is propagated to the failure terminal of a Timeout Control node" on page 73
- "Message processing fails within a Timeout Notification node" on page 73
- "An MRM CWF message is propagated to the failure terminal" on page 73
- "An MRM XML message exhibits unexpected behavior" on page 73
- "You encounter problems when messages contain EBCDIC New Line characters" on page 74

- "The MIME parser produces a runtime error while parsing a message" on page 75
- "You get runtime errors writing a MIME message from the logical message tree" on page 75
- "Error message BIP5651 is issued when receiving a SOAP with Attachments message from a WebSphere Application Server client" on page 75
- "WebSphere Application Server produces an error when receiving a SOAP with Attachments message" on page 76
- "Unexpected characters are displayed when using the XMLTransformation node on z/OS" on page 77
- "BIP5117 The MRM parser has failed to parse a message because two attributes have the same name"
- "Problems when using code page translation on HP-UX" on page 77

BIP5117 The MRM parser has failed to parse a message because two attributes have the same name

- **Scenario:** Two attributes in different name spaces have identical names.
- **Explanation:** The MRM (Xerces) parser has failed to parse the message.
- **Solution:** Modify the attribute names so that they are not identical. This is a know problem with the Xerces parser.

A communication error is issued when you use the enqueue facility

- **Scenario:** You use the enqueue or dequeue tools to put a message on a queue, but an error message is issued indicating that there is a communication error with the queue manager name.
- **Explanation:** The WebSphere MQ queue manager has not started.
- **Solution:** Restart the WebSphere MQ queue manager.

The enqueue facility is not picking up changes made to a message

- **Scenario:** You are using the Message Brokers Toolkit's message enqueue facility to put messages to WebSphere MQ queues. You have updated a message and want to put the message to the queue, but your changes do not seem to have been picked up.
- **Solution:**
  1. Close and then reopen your enqueue file.
  2. Select the message that you want to put to the queue.
  3. Save and close the enqueue file.
  4. Select the menu next to the **Put a message to a queue** icon.
  5. Click **Put message**.
  6. Click the enqueue file in the menu.
  7. Click **Finish**.

  This puts your updated message to the queue.

You do not know which header elements have any effect in enqueue

- **Scenario:** When using the Enqueue editor, the accounting token, correlation ID, group ID, and message ID in the message header do not seem to have any effect.
- **Explanation:** These fields are not properly serialized so they have no effect.

Enqueue message files are still listed after they have been deleted

- **Scenario:** Enqueue message files are still listed in the drop-down menu after they have been deleted.
- **Explanation:** Deleted enqueue files are not removed from the drop-down menu. Selecting these files has no effect.

The ESQL transform of an XML message gives unexpected results

- **Scenario:** You have created an XML message that looks like this:

  ```
  <?xml version ="1.0" standalone="no"?><!DOCTYPE doc
  [<!ELEMENT doc (#PCDATA)*>]><doc><I1>100</I1></doc>
  ```

  You apply the ESQL transform:

  ```
  SET OutputRoot.XML.doc.I1 = 112233;
  ```

  This generates the XML message (after serialization):

  ```
  <?xml version ="1.0" standalone="no"?><!DOCTYPE doc
  [<!ELEMENT doc (#PCDATA)*>]<I1>112233<I1>><doc><I1>100</I1></doc>
  ```

  The new value for *I1* has been put inside the DOCTYPE, and has not replaced the value of *100*, as you expected.
- **Explanation:** The XML in your message contains two doc elements:
  - The doctype element
  - The xmlElement that represents the body of the message

  The parser has found the first instance of an element called doc and has created a child *I1* with the value *112233*.
- **Solution:** To assign a new value to the element *I1* within the message body element doc, explicitly identify the second doc element, like this:

  ```
  SET OutputRoot.XML.(XML.tag)doc.I1 = 112233;
  ```

Error message BIP5005 is issued by the Compute node

- **Scenario:** You send a simple XML message into a simple message flow. The message is:

  ```
  <doc><I1>100</I1></doc>
  ```

  The Compute node in the message flow contains the following ESQL:

  ```
  SET OutputRoot.XML.abc = InputBody;
  ```

  You expect the following output message to be created:

  ```
  <abc><doc><I1>100</I1></doc></abc>
  ```

  The Compute node generates error message BIP5005 and does not implement the ESQL.
- **Explanation:** You are assigning an element of one type (root) to an element of another type (xmlElement). The parser does not do this implicit cast for you.
- **Solution:** You can do the cast yourself in the ESQL to achieve the result that you want, using either of the following two casts:

  ```
  SET OutputRoot.XML.(XML.Element)abc = InputBody;
  ```

  or:

  ```
  SET OutputRoot.XML.(XML.tag)abc = InputBody;
  ```

A message is propagated to the failure terminal of a Timeout Control node

- **Scenario:** The Timeout Control node receives a message with invalid, corrupt or missing timeout information. The message is propagated to the failure terminal of the Timeout Control node and an exception list is generated.
- **Explanation:** The following error conditions can cause propagation to the failure terminal:
  - A timeout request has no Action or no Identifier.
  - A SET request has an Identifier that matches an existing stored SET request for this Timeout Control node (identified by the Unique Identifier property) and AllowOverwrite of the original request is set to FALSE.
  - A CANCEL request has an Identifier that does not match an existing stored SET request for this Timeout Control node (identified by the Unique Identifier property).
  - A SET request has a Count of 0 or is less than -1.
  - The StartDate or StartTime are not in the correct format (or one of the understood keywords).
  - The calculated timeout is in the past.
  - The Interval is less than 0, or 0 with a Count of -1.
- **Solution:** Check for any of the error conditions listed above and rectify them.

Message processing fails within a Timeout Notification node

- **Scenario:** A message is propagated to the failure or catch terminal of a Timeout Notification node.
- **Explanation:** If there is an error in the processing of a timeout within the Time Notification node, an exception list is generated and a message is propagated to the failure terminal. This is done under the same transaction, if one is being used. If the failure terminal is not wired, propagation does not occur.

  If an error happens downstream of the Timeout Notification node after a successful propagation (either to the out or failure terminal), the message is propagated to the catch terminal (all under the same transaction). If the catch terminal is not wired, or the propagation along the catch node fails, the processing of that timeout is rolled back.
- **Solution:** Ensure that the failure and catch terminals of your Timeout Notification node are wired correctly.

An MRM CWF message is propagated to the failure terminal

- **Scenario:** Your MRM CWF message is propagated to a failure terminal, and generates error messages BIP5285, BIP5125, and BIP5181 or messages BIP5285, BIP5125, and BIP5288.
- **Explanation:** These errors report an inconsistency between the length of the message being processed, and the length of the message that is defined in the message model.
- **Solution:** Ensure that the length of the message as defined in the CWF layer is accurate. Check and correct the definition.

An MRM XML message exhibits unexpected behavior

- **Scenario:** Your message flow is processing a message that you have modelled in the MRM. The message tree has not been created as you expected, the output XML message does not have the expected contents, or the message contents are not being validated. No error message has been issued.
- **Explanation:** There are two possible reasons for this:

- **Explanation 1:** The XML physical format settings for a message set contain a property called Root Tag Name. This defaults to MRM, in order to remain compatible with previous releases of the product. If you have not deleted the contents of this field, the MRM XML parser expects the root tag for all XML messages to be MRM.

  Solution 1: Clear this field, or set it to the root tag used by all your XML messages. If you provide a value in this field, the root tag does not need to be modelled in all your message definitions.

- **Explanation 2:** In order to remain backwards compatible, the broker recognizes the format XML and invokes the XML parser with specific default values. If you have created an XML physical layer for this message with the name XML, the broker uses your definition. However, if you have not created an XML physical layer with this name, but have specified XML as the format, either in the input node or the MQRFH2 header (when the input bit stream is parsed to a message tree), the broker accepts the value specified and passes default values to the parser to create the message tree.

  Similarly, if you set XML in the Properties folder for the output message in the Compute node, this value is passed to the parser when it creates the message bit stream from the message tree, usually in the Output node.

  The use of these default values by the parser might result in different content, structure, or both, for either message tree or output message. You can find further information about the action taken by the broker in the user trace log where the following information might be written:

```
XMLWorker::initializeParse file:C:\s000\src\cpi\pwf\xml\xmlworker.cpp
line:126 message:5409.BIPv600
No dictionary present have you specified Wire Format 'XML' in error? ,
UserTrace   BIP5409E: XML Worker: Wire Format 'XML' specified.
                      Default MRM XML settings are being used because wire format
                      identifier 'XML' was specified and not found.
                      This can be due to an incorrect setting of the wire format
                      identifier in a message.
```

  Solution 2: If you have incorrectly entered the identifier of the format that you have defined, correct your code and retry. If you do not want the default action to be taken, define a physical layer that produces the required results.

You encounter problems when messages contain EBCDIC New Line characters

- **Scenario:** If your bit stream input message contains EBCDIC New Line (NL) characters, problems might arise if your message flow changes the target CCSID to an ASCII CCSID. For example, during conversion from CCSID 1047 (EBCDIC used for z/OS Open Edition) to CCSID 437 (US PC ASCII), an NL character is translated from hex ’15’ to hex ’7F’, which is an undefined character. This is because there is no corresponding code point for the New Line character in the ASCII code page.
- **Solution:** You can overcome the problem in these cases:
  - On a system where the queue manager uses an ASCII codeset, you can make sure that incoming messages do not contain any EBCDIC NL characters by:
    - Specifying that WebSphere MQ performs the conversion at the input node
    - Setting the queue manager attribute to convert NL to Line Feed (LF)
  - Where the input bit stream is character data, you can use MRM tagged/delimited message sets in a Compute node to replace the NL character with the desired output.

The MIME parser produces a runtime error while parsing a message

- **Scenario:** A MIME message is received by a message flow and produces a runtime error when the message is parsed.
- **Explanation:** The following errors can cause the MIME parser to reject a message:
  - The MIME header is not correctly formatted.
  - Either the top-level MIME header block, or a MIME header block for a nested multipart part, does not have a valid Content-Type header.
  - The top-level Content-Type has a media type of `message`.
  - The top-level Content-Type has a media type of `multipart` and no boundary definition.
  - A MIME header block is not correctly terminated by a blank line.
  - The constituent MIME parts are not correctly separated by boundary lines.
  - A boundary parameter value occurs in the content of a MIME part.
- **Solution:** Check the MIME message for any of the error conditions listed above and rectify them.

You get runtime errors writing a MIME message from the logical message tree

- **Scenario:** You are writing a MIME logical message tree out as a bit stream and the parser generates a runtime error.
- **Explanation:** The following errors can cause the MIME parser to reject a logical message tree:
  - The root of the tree is not called MIME.
  - The last child of MIME is not called Parts or Data.
  - Parts has a value-only element and this element is not the first or last child of Parts.
  - Parts has children that are not value-only elements or Part children.
  - Parts does not have any children called Part.
  - The last child of a Data element is not a BLOB.
- **Solution:** Check the MIME logical message tree for any of the error conditions listed above and rectify them.

Error message BIP5651 is issued when receiving a SOAP with Attachments message from a WebSphere Application Server client

- **Scenario:** When a WebSphere Application Server client sends a SOAP with Attachments message to the broker over JMS, error message BIP5651 is issued stating that no valid Content-Type header has been found.
- **Explanation:** When a WebSphere Application Server client sends a SOAP with Attachments message to the broker over JMS, the MIME Content-Type value appears in the RFH2 header and not in the MIME message body.
- **Solution:** To solve this problem, the Content-Type value needs to be copied from the RFH2 header to the beginning of the message as a MIME header before the message is parsed. The following ESQL adds the Content-Type value to the beginning of the WebSphere Application Server message and then invokes the MIME parser on the result.

```
create procedure parseWAS_JMS(IN InputMessage reference,IN OutputMessage reference)
/***********************************************************************
* convert a WAS/JMS message to the correct format for the MIME parser
```

```
***********************************************************************/
begin
  -- get the data as a BLOB
  declare body BLOB InputMessage.BLOB.BLOB;

  -- get the Content-Type value from the RFH2 header. Content-Type is the only
  -- header which is critical for the MIME parser, but the same approach can be
  -- used for any MIME headers which have been stored under the RFH2 header.
  declare contentType char InputMessage.MQRFH2.usr.contentType;

  -- add the contentType to the bitstream as part of an RFC822 header block
  set body = cast(('Content-Type: '||contentType) as blob ccsid 819)||x'0d0a0d0a'||body;

  -- invoke MIME parser on the modified bitstream
  CREATE LASTCHILD OF OutputMessage DOMAIN('MIME') PARSE(body);
end;
```

A message flow can take in the JMS message in the BLOB domain and then invoke the procedure above from a Compute node. The procedure can be called using the following ESQL from a Compute node:

```
 CALL CopyMessageHeaders();              -- standard procedure to copy headers
 CALL parseWAS_JMS(InputRoot, OutputRoot);  -- parse the 'body' as MIME
```

WebSphere Application Server produces an error when receiving a SOAP with Attachments message

- **Scenario:**When sending a SOAP with Attachments message to a WebSphere Application Server client using JMS an error is generated stating that the bit stream contains unexpected characters.
- **Solution:** When the broker sends a SOAP with Attachments message to WebSphere Application Server over JMS, the MIME Content-Type value must appear in the RFH2 header and not in the body of the message. The following procedure removes any MIME style headers from the front of the message bit stream and adds the Content-Type value to the RFH2 header. The supplied boundary value allows you to locate the start of the multipart MIME content.

```
create procedure writeWAS_JMS(IN OutputTree reference,IN boundary char)
/****************************************************************************
* Serialise a MIME tree as normal, but then strip off any initial headers
* and save the Content-Type value in the RFH2 header as expected by WAS/JMS.
* Note: boundary - must be supplied with the leading hyphen pair
****************************************************************************/
begin
  -- convert MIME subtree to BLOB
  declare body BLOB asbitstream(OutputTree.MIME);

  -- locate first occurrence of boundary and discard any data before this
  declare firstBoundary integer;
  set firstBoundary = position (cast(boundary as blob ccsid 819) in body);
  set body = substring(body from firstBoundary);

  -- save the MIME Content-Type value in the RFH2 header. Any other MIME
  -- headers which need to be preserved in the RFH2 header can be handled
  -- in the same way
  set OutputTree.MQRFH2.usr."contentType" = OutputTree.MIME."Content-Type";

  -- clear the MIME tree and create a new BLOB child for the modified body
  set OutputTree.MIME = null;
  CREATE LASTCHILD OF OutputTree DOMAIN('BLOB')PARSE(body);
end
```

Before calling this procedure, the message flow needs to be able to obtain the value of the boundary. This value might only be available within a Content-type header. The following procedure allows you to extract the Boundary value:

```
create procedure getBoundary(IN ct reference,OUT boundary char)
/****************************************************************
* return value of the boundary parameter from a Content-Type value
****************************************************************/
begin
  declare boundaryStart  integer;
  declare boundaryEnd    integer;

  set boundaryStart  = position('boundary=' in ct) + 9;
  set boundaryEnd    = position(';' in ct from boundaryStart);
  if (boundaryStart <> 0) then
    if (boundaryEnd <> 0) then
     set boundary  = substring(ct from boundaryStart for boundaryEnd-boundaryStart);
    else
      set boundary  = substring(ct from boundaryStart);
    end if;
  end if;
end;
```

A Compute node can invoke these procedures for sending a MIME message using the following ESQL:

```
SET OutputRoot = InputRoot;

declare boundary char;
CALL getBoundary(OutputRoot.Properties.ContentType, boundary);

CALL writeWAS_JMS(OutputRoot,boundary);
```

Unexpected characters are displayed when using the XMLTransformation node on z/OS

- **Scenario:** When using the XMLTransformation node on z/OS, all the uppercase Os that are in an XML file on the incoming message are changed to underscore characters.
- **Explanation:** The XMLTransformation node input message must come in as ASCII for the transform to work correctly. The XMLTransformation node does not work with XML or XSL data in EBCDIC code. Java assumes a conversion from EBCDIC 1047. WebSphere Message Broker then converts to EBCDIC 500, because the CCSID is set to 500. EBCDIC 1047 and EBCDIC 500 are very similar. Only uppercase O, J and Z are different. (J and Z are also converted incorrectly). The conversions leave a string that is unreadable because it is really in ASCII. However, it does convert the O from an EBCDIC 1047 character to an EBCDIC 500 character.
- **Solution:** Change your program either to do a string assignment without any conversions, or specify that the string is in ASCII ISO-8859-1 (CCSID 819).

Problems when using code page translation on HP-UX

- **Scenario:** You experience code page translation problems on HP-UX.
- **Solution:** Check the WebSphere MQ queue manager attribute *CodedCharSetID*. The default value for this attribute is 1051. Change this to 819 for queue managers that host WebSphere Message Broker components.

Resolving problems when using the workbench

This topic contains advice for dealing with some common problems that can arise when using the workbench.

Connecting:

- “Your broker is not recognized by the workbench” on page 78

- “You cannot connect to your Configuration Manager from the workbench”

Error messages:

- “Errors occur while using the workbench” on page 79
- “An error is issued when you access the help system” on page 79
- “The Message Brokers Toolkit displays an error message after the error has been fixed” on page 79
- “Error message BIP0800E is issued when you rename a domain object” on page 79
- “Error message BIP0935E is issued when you edit domain connection parameters” on page 80
- “Error message BIP0876W is issued when you make changes in the Broker Topology editor” on page 80
- “The message ″Unable to create part: filename.extension″ is issued” on page 80

Appearance:

- “You cannot find the Perform button in a cheat sheet” on page 80
- “You cannot close a message dialog” on page 81
- “The Domains navigator content is updated but you have not made any changes” on page 81
- “A server project containing a valid domain connection file is not displayed” on page 81
- “Some objects are not visible in the Domains Navigator” on page 81
- “You cannot find what you are looking for in the Tasks view” on page 81
- “Your message flow and message set projects look different” on page 81

Other:

- “Deleting or closing a project takes a long time” on page 82
- “You are thinking of moving RDB artefacts within the workbench” on page 82
- “You do not know how to return to the welcome page” on page 82

Your broker is not recognized by the workbench

- **Scenario:** Your broker is not recognized by the workbench.
- **Explanation:** Broker names in the workbench are case sensitive.
- **Solution:** When you identify a broker in the workbench, make sure that you use the correct case for its name. If it was created with an uppercase name, you must specify the name in uppercase in the workbench. The broker’s name when it was created might have become uppercase due to case-folding, even though you entered its name in lowercase.

You cannot connect to your Configuration Manager from the workbench

- **Scenario:** You have problems when trying to connect to your Configuration Manager from the workbench.
- **Solution:** Perform the following checks when connecting to the Configuration Manager from the workbench:

  - Is the Configuration Manager running? Check the system event log to ensure that your Configuration Manager is available for use, and take action to correct any errors shown.
  - Is the listener on the Configuration Manager's queue manager running, and listening on the correct port?
  - Are you a member of the correct security groups? Check that you belong to the required user group or ACL group.

Errors occur while using the workbench

- **Scenario:** An error has occurred while you are using the workbench, and you want information to help you to diagnose the problem.
- **Solution:** There are a number of places where you can find information that might help you to determine the cause of your problem. Some errors that occur in the workbench are logged in the .log file in your workspace\.metadata directory. You can view this log by switching to the Plug-in Development perspective, and in the Tasks view, in the bottom right pane, clicking the Error Log tab. This shows in which plug-in the error occurred, and gives further information.

  You can also use trace to try and determine the cause of your problem. See "Using trace" on page 109 for more information about tracing.

An error is issued when you access the help system

- **Scenario:** You are accessing the help system through the workbench, and an error message is issued indicating that the web page that you requested is not available offline.
- **Explanation:** This error might occur if you previously had a connection to a Web-based version of the help system and lost it, or if you have **Work Offline** selected in your Internet Explorer options.
- **Solution:** Click **Connect** to enable the help system to load.

The Message Brokers Toolkit displays an error message after the error has been fixed

- **Scenario:** The Message Brokers Toolkit is in an inconsistent state or displays an error message after the error has been fixed
- **Solution:** Clean the workspace by performing the following steps:
  1. From the Message Brokers Toolkit, click **Project** → **Clean**.
  2. Click **Clean all projects** and **Finish**.

     This cleans the whole workspace of any internal files, which are then re-created so that none of your data is lost.

Error message BIP0800E is issued when you rename a domain object

- **Scenario:** Error message BIP0800E is issued when you click **OK** in the rename dialog on a domain object.
- **Explanation:** This error occurs when the Domains Navigator receives a notification about object creation or deletion and the Navigator changes its selection while you are entering the new name in the dialog.
- **Solution:** Refresh the parent of the object that you are trying to rename to see whether the action was sent to the Configuration Manager before the error occurred. If the name of the object has not changed, wait for an automatic refresh in the Domains Navigator and try to rename the object again. An automatic refresh typically occurs when you or another user have saved many changes in the Topics hierarchy.

Error message BIP0935E is issued when you edit domain connection parameters

- **Scenario:** You want to edit the connection parameters for the domain, and see only one connection file. However, error message BIP0935E is issued indicating that several domain connection files exist for the domain.
- **Explanation:** This error can occur when you delete the port value in the Domain Connection editor and save this file. The corresponding domain is displayed in the Domains view with a label like `yourQMGR@yourMachine:`. Error BIP0935E is issued when you try to edit parameters for this domain.
- **Solution:** In the Broker Administration perspective, edit the corresponding .configmgr file to set the port to 1414, and save the file. Even if the domain label is not updated in the Domains view, the port parameter has been taken into account in the domain connection file.

  If you want to see the correct label and open the Domain Connection editor from the Domains view, close and restart the workbench.

Error message BIP0876W is issued when you make changes in the Broker Topology editor

- **Scenario:** While you are making some changes in the Broker Topology editor, error message BIP0876W is issued and you are invited to close the editor without saving.
- **Explanation:** The editor has been told that Configuration Manager content has changed, and the changes that you have initiated will conflict with those recent changes. This is typically because another user has made some changes on the topology. To avoid potential conflicts between other users' changes and yours, you are invited to close the editor without saving. You can then reopen the editor, which will be populated with the actual Configuration Manager content.
- **Solution:** If you are sure that your changes will not be in conflict with those made by the other user and you want to save your changes, you can save. The Configuration Manager tries to apply your changes on top of its actual content. You are notified if some of the requests fail.

  The same message might be raised in the Topics editor for the same reason.

The message ″Unable to create part: filename.extension″ is issued

- **Scenario:** You open an editor in the workbench and an error message is raised indicating that a file cannot be created.
- **Explanation:** There are two possible explanations for this problem:
  - **Explanation 1:** The given file is not associated with any editor.

    Solution 1: Right-click the file and choose the default editor to open this file, or choose any other editor if the default editor cannot open this file.
  - **Explanation 2:** The given file contains syntax errors and cannot be loaded into the chosen editor. However, if you then try and open a valid file, you get the same error message over and over again.

    Solution 2: Restart the workbench to resolve this problem, and do not try to load the file into an editor again until you have fixed the syntax errors.

You cannot find the Perform button in a cheat sheet

- **Scenario:** You are attempting to use the Preparing Samples for First Use cheat sheet in the workbench. The instructions indicate that you need to click **Perform**, but you cannot find this button.
- **Solution:** The **Perform** and **Skip** buttons are displayed only after you have clicked **Click to Begin**. This can be found at the bottom of the introduction

section of the cheat sheet. After you have clicked **Click to Begin**, the **Perform** and **Skip** buttons are displayed at the bottom of each section.

You cannot close a message dialog

- **Scenario:** You have opened a pop-up menu, and a message dialog box appears to tell you that previous changes that you made have been processed successfully by the Configuration Manager. You cannot close the dialog.
- **Solution:** If you cannot close the dialog by clicking **OK**, press **Escape**.

The Domains navigator content is updated but you have not made any changes

- **Scenario:** You are connected to a domain and have not performed or initiated any changes. However, the Domains Navigator and editor's content are updated.
- **Explanation:** Another user has made changes on the Configuration Manager. The Domains Navigator and editors are notified of these changes and are refreshed automatically.

A server project containing a valid domain connection file is not displayed

- **Scenario:** You have a valid domain connection file in one of your workspace server projects, but you cannot see this project in the Broker Administration perspective, or see an associated domain in the Domains Navigator.
- **Explanation:** The project is closed. Closed projects are not displayed in the Broker Administration perspective, and their content is not analyzed.
- **Solution:** Open the project from the Resource Perspective Navigator. The project then appears in the Broker Administration perspective, and its content is analyzed and displayed. The domain connection and domain then appear in the appropriate navigators.

Some objects are not visible in the Domains Navigator

- **Scenario:** Some objects, for example broker topologies, topics, brokers or execution groups, are not visible in the Domains Navigator.
- **Explanation:** ACL-restricted objects might not be visible in the Domains Navigator if the corresponding **Show ACL restricted objects in Navigators** preference was previously not selected.
- **Solution:**
  1. Click **Window** → **Preferences** → **Broker Administration preference page**.
  2. Check whether the **Show ACL restricted objects in Navigators** preference is selected. If it is not, select this preference and click **OK**.

     Any restricted objects will be displayed in the Domains Navigator.

You cannot find what you are looking for in the Tasks view

- **Scenario:** The Tasks view has such a large number of entries that it is difficult to find the entry that you want.
- **Solution:** At the top of the Tasks View, click the icon with the arrows pointing to the right. This launches the Filter Tasks dialog box. You can tailor your selections to display only a subset of the task list entries. One particularly useful filter is to show only those task list entries for the currently selected project.

Your message flow and message set projects look different

- **Scenario:** You have created a new simple project in the workbench and now all your message flow and message set projects look different.

- **Explanation:** When you create a new simple project, the workbench switches automatically to the Resource Navigator view.
- **Solution:** To return to the previous view, switch to the Broker Application Development perspective.

Deleting or closing a project takes a long time

- **Scenario:** Deleting or closing a project to save memory takes a long time.
- **Explanation:** If a project is referenced by other projects, removing that project requires all the other projects, and the projects that refer to them recursively, to be built fully. This keeps the tool internal content-assist and validation models current.
- **Solution:** There is very little overhead in keeping a project open in the workspace, so it is recommended that you do not close or delete projects that you need.

You are thinking of moving RDB artefacts within the workbench

- **Scenario:** You are considering moving RDB artefacts, such as the XMI files for connection, database, schema and tables, within the workbench.
- **Solution:** You cannot move the RDB artefacts once they are imported. The only move that you can perform without affecting the integrity of the files is a move between two projects while keeping the project-relative path of the files unchanged. If you want to place the RDB artefacts in a different folder, import them again from the database.

You do not know how to return to the welcome page

- **Scenario:** You do not know how to return to the welcome page that was displayed in the workbench when you first started using it.
- **Solution:** To open the welcome page:
  1. From the **Help**menu, select **Welcome**. This displays a list of the welcome pages available.
  2. Select the welcome page that you want, for example, **Message Broker Toolkit for WebSphere Studio - Message Broker**.
  3. Click **OK**.

Resolving problems when using databases

This topic contains advice for dealing with some common problems that can arise when using databases:

- “DB2 error message SQL0805N is issued” on page 83
- “DB2 error message SQL0998N is issued” on page 83
- “DB2 error message SQL0998N or SQL1248N is issued” on page 84
- “DB2 error message SQL1040N is issued” on page 84
- “DB2 error message SQL1224N is issued when you connect to DB2” on page 84
- “DB2 or ODBC error messages are issued on z/OS” on page 86
- “You do not know how many database connections a broker requires” on page 86
- “You want to use XA with DB2” on page 87
- “XA coordination fails with DB2 V8 Fix Pack 2” on page 87
- “Error message BIP2322 is issued when you access DB2 on z/OS” on page 87
- “On Oracle, a database operation fails to return any rows, even though the rows exist” on page 87

- "There is a memory leak from the Oracle Client Interface on HP-UX 11" on page 88
- "Stored procedure parameter sizes are different in Oracle 9i and Oracle 10g" on page 88
- "Error message BIP2731 is issued when you use retained publications with a Sybase database" on page 88
- "Database updates are not committed as expected" on page 89
- "The DataDirect trace system is unable to open the ODBC trace file" on page 89
- "You want to list the database connections that the broker holds" on page 89

DB2 error message SQL0805N is issued

- **Scenario:** When you run the **mqsicreatebroker** command, or when a message flow that includes a Database node runs, SQL error `SQL0805N NULLID.SQLLF000` is issued.
- **Solution:** Open a DB2 Command Line Processor window and issue a bind command to the database.

  On Linux and UNIX systems, enter the commands:

  ```
  connect to db
  bind ~/sqllib/bnd/@db2cli.lst grant public CLIPKG 5
  connect reset
  ```

  where *db* is the database name.

  On Windows systems, enter the commands:

  ```
  connect to db
  bind x:\sqllib\bnd\@db2cli.lst blocking all grant public
  connect reset
  ```

  where *x:* identifies the drive onto which you installed DB2, and *db* is the database name.

DB2 error message SQL0998N is issued

- **Scenario:** When you use a globally coordinated message flow with a DB2 8.1 database, error SQL0998N is issued with Reason Code 09 and Subcode 02.
- **Solution:** Use the following procedure to set the DB2 database manager configuration parameter **TP_MON_NAME** to *MQ*, and alter the XAOpenString in the qm.ini file for your queue manager:
  1. To set the **TP_MON_NAME** parameter, use the following command from a login that has administrative authority for the DB2 instance:

     ```
     db2 update dbm cfg using TP_MON_NAME MQ
     ```

  2. Stop and restart the DB2 instance so that the value is recognized.
  3. To set the AXLIB value in the qm.ini file, alter the XAOpenString to match the following:

     ```
     XAOpenString=DB=databasename,UID=userid,PWD=password,AXLIB=library
     ```

  4. Set the AXLIB library path to one of the following values depending on your operating system:
     - On Windows:

       ```
       AXLIB=mqmax
       ```

     - On AIX:

       ```
       AXLIB=usr/mqm/lib/libmqmax_r.a
       ```

     - On HP-UX:

       ```
       AXLIB=/opt/mqm/lib/libmqmax_r.sl
       ```

     - On Linux:

       ```
       AXLIB=/opt/mqm/lib/libmqmax_r.so
       ```

- On Solaris:

  ```
  AXLIB=/opt/mqm/lib/libmqmax.so
  ```

5. Stop and restart your broker and queue manager for these changes to take effect.

DB2 error message SQL0998N or SQL1248N is issued

- **Scenario:** When you try to use a globally coordinated message flow with a DB2 version 8.1 database, you get either of the following error messages:
  - SQL0998N with Reason Code 09 and Subcode ″″
  - SQL1248N with the message ″database not defined with the transaction manager″
- **Solution:** Use the instructions in Configuring databases for coordinated message flows using DB2 to configure the database and XAResourceManager stanza.

DB2 error message SQL1040N is issued

- **Scenario:** You are using DB2, and error message BIP2322 is issued with error SQL1040N.
- **Explanation:** The following DB2 message indicates that the value of the DB2 database configuration parameter **maxappls** has been reached:

  ```
  "SQL1040N The maximum number of applications is already connected to the database.
  SQLSTATE=57030"
  ```

  DB2 has rejected the attempt to connect.

  If this database is one of the defined broker databases, implying that a broker thread connection request has failed, the broker is probably not functioning correctly.
- **Solution:**
  1. Stop all brokers that connect to the affected database.
  2. Increase the value of the **maxappls** configuration parameter. Also, check the value of the associated parameter **maxagents** and increase it in line with **maxappls**.
  3. Restart the DB2 database.

DB2 error message SQL1224N is issued when you connect to DB2

- **Scenario:**DB2 error message SQL1224N is issued when you connect to DB2, indicating that a database agent could not be started or was terminated as a result of a database shutdown or force command.
- **Solution:** On AIX, use the TCP/IP node to connect to DB2 to avoid the limit of 10 connections. To set up AIX and DB2 loop-back to use the TCP/IP node:
  1. Use the **mqsideletebroker** command to delete your broker.
  2. Configure DB2 to use TCP/IP, and to start the TCP/IP listener. On the server machine, log in as the DB2 instance owner, usually `db2inst1`, and issue the commands:

     ```
     db2set DB2COMM=tcpip
     db2stop
     db2start
     ```

  3. If the DB2 connection port is not defined in /etc/services, edit the services file to add the DB2 connection and interrupt ports. You must use unique names, and port numbers that are not already defined in the services file; for example:

     ```
     db2svc1     3700/tcp          # DB2 Connection Service
     db2isvc1    3701/tcp          # DB2 Interrupt Service
     ```

4. Update the DB2 configuration; for example:

   ```
   db2 update dbm cfg using svcenamedb2svc1
   ```

   where *db2svc1* is the name of the DB2 Connection port service in /etc/services.

   Alternatively, you can specify a port number directly.

5. Stop and restart the database using the commands:

   ```
   db2stop
   db2start
   ```

6. Catalog a new TCP/IP node:

   ```
   db2 catalog tcpip node NODENAME remote HOSTNAME server db2svc1
   ```

   where:

   NODENAME
   : is the name of the new node. You can use local as your node name, providing it is unique.

   HOSTNAME
   : is the name of your computer.

   db2svc1
   : is the name of the DB2 connection port service in `/etc/services`.

   Message DB20000I is displayed when the command completes successfully.

7. Catalog the database with a new alias name; for example:

   ```
   db2 catalog database DATABASE as DBALIAS at node NODENAME
   ```

   where:

   DATABASE
   : is the physical name of the database.

   DBALIAS
   : is the database alias name that you want to use.

   Specify the new alias name in all subsequent references to the local database, for example when you run the **mqsicreatebroker** command.

8. Stop and start DB2.

   ```
   db2 terminate
   db2stop
   db2start
   ```

9. Log on using the broker service user ID.

10. To enable WebSphere Message Broker brokers to use the database, update the ODBC configuration file to add definitions for the database (if you are using the default odbc.ini file, this is located at /var/wmqi/odbc/.odbc.ini):

    a. At the top of the file add a definition for the database alias name, as follows:

       ```
       DBALIAS=IBM DB2 ODBC Driver
       ```

    b. Add a new stanza for the database alias:

       ```
        [DBALIAS]
       Driver=INSTHOME/sqllib/lib/libdb2.a
       Description=Broker Database Alias
       Database=DBALIAS
       ```

       where *INSTHOME* is the path to your DB2 Instance directory.

11. Create a new broker using the `DBALIAS` on the **-n** flag, as follows:

```
mqsicreatebroker BrokerName -i mqsiuid -a mqsipw -q BrokerQueueManager -n DBALIAS -n
```

12. Start the broker and redefine it in the workbench.
13. Deploy the broker and test the flows.

DB2 or ODBC error messages are issued on z/OS

- **Scenario:**DB2 or ODBC messages are issued on z/OS indicating that:
  - An exception was caught while issuing the database SQL connect command.
  - There was a database error with an ODBC return code of -1, an SQL state of 58004 and a native error code of -99999.
  - The broker made an unsuccessful attempt to access its database using a particular user ID.
- **Solution:** If an ODBC message is displayed:
  1. Turn ODBC application tracing on to produce the traceodbc file.
  2. Locate the traceodbc file. This is in the /output subdirectory. For example /u/argo/VCP0BRK/output/traceodbc.
  3. Go to the bottom of this file and search for previous instances of SQLerror.

  Common DB2 problems include:
  - ODBC return code -1, SQL state 58004, Native error code -99999

    This might be caused by:
    - No SQL code. The DB2 subsystem is not started
    - RRS is not started.
  - SQLCODE 922.

    The user ID of the started task is not authorized to use plan DSNACLI.
  - ODBC return code -1, SLQ state 42503, Native error code -553

    This might be caused by the user ID of the started task not being authorized to use the current SQL ID. Re-configure the broker and specify `DB2_TABLE_NAME` as a valid name, or create an RACF group and connect the started task user ID to this group.

You do not know how many database connections a broker requires

- **Scenario:** You do not know how many database connections to set up for your broker.
- **Solution:** Determine the number of database connections required by a broker for capacity and resource planning. On DB2, the default action taken is to limit the number of concurrent connections to a database to the value of the **maxappls** configuration parameter; the default for **maxappls** is 40. The associated parameter **maxagents** also impacts the current connections.

  The connection requirements for a single message broker are:
  - Five are required by internal broker threads.
  - One is required for each publish/subscribe neighbor, if the topology has been deployed.
  - One is required for each message flow thread that contains a Publication node.
  - One is required for each message flow thread that parses MRM messages.
  - One is required for each database access node to separate ODBC data source names for each message flow thread (that is, if the same DSN is used by a different node, the same connection is used).

  - If you are using SCADA nodes with WebSphere MQ Everyplace, add a further number of connections depending on whether thread pooling is being used (check the Use Thread Pooling property of the SCADAInput node):
    - If Use Thread Pooling is not selected (the default), add the number of SCADA clients that will connect to the SCADAInput node.
    - If Use Thread Pooling is selected, add the value in the Max Threads property of the SCADAInput node. The default value is 500.

  Connections for neighbors and publication nodes are required only if you are using retained publications.

You want to use XA with DB2

- **Scenario:** You want to use XA with DB2.
- **Solution:** If you want to use XA coordination with DB2 V8.1, ensure that your queue manager is configured to use **ThreadOfControl=THREAD**. You can configure this parameter in the XAResourceManager stanza in qm.ini on Linux or UNIX, or use WebSphere MQ Services on Windows.

XA coordination fails with DB2 V8 Fix Pack 2

- **Scenario:** XA coordination fails with DB2 V8 Fix Pack 2 on AIX, HP-UX, Linux, Solaris, or Windows.
- **Explanation:** If a non-XA coordinated message flow tries to access a database that was previously being used by an XA coordinated message flow, failure can occur because DB2 believes that the later flow is still coordinated from the previous flow.
- **Solution:** This is fixed in DB2 V8 Fix Pack 3 by APAR IY44711.

Error message BIP2322 is issued when you access DB2 on z/OS

- **Scenario:** You are using a message flow in which a Database node, Compute node, or Filter node attempts to access a table on a DB2 data-sharing group other than the one that the broker is using. If ODBC tracing is turned on, an error message is written to the traceodbc file:

  ```
  SQLError( hEnv=0, hDbc=0, hStmt=1, pszSqlState=&302f8ecc, pfNativeError=&302f8ec8,
  pszErrorMsg=&28f6a6d0, cbErrorMsgMax=1024, pcbErrorMsg=&302f8eb4 )
  SQLError( pszSqlState="51002", pfNativeError=-805, pszErrorMsg="{DB2 for OS/390}
  {ODBC Driver}{DSN06011}
   DSNT408I SQLCODE = -805, ERROR:  DBRM OR PACKAGE NAME DSN610GH..DSNCLICS.16877-
            BE5086005F4 NOT FOUND IN PLAN DSNACLI. REASON 02
   DSNT418I SQLSTATE   = 51002 SQLSTATE RETURN CODE
   DSNT415I SQLERRP    = DSNXEPM SQL PROCEDURE DETECTING ERROR
   DSNT416I SQLERRD    = -350  0  0  -1  0  0 SQL DIAGNOSTIC INFORMATION
   DSNT416I SQLERRD    = X'FFFFFEA2'  X'00000000'  X'00000000'  X'FFFFFFFF'
            X'00000000'  X'00000000' SQL DIAGNOSTIC INFORMATION
  ```

  This error is accompanied by the following BIP2322E message in the syslog:

  ```
  BIP2322E: DATABASE ERROR: SQL STATE '51002'; NATIVE ERROR CODE '-805'
  ```

- **Explanation:** This happens when the DSNACLI plan has not been bound in the correct way.
- **Solution:** Ensure that the DSNACLI plan is bound correctly.

On Oracle, a database operation fails to return any rows, even though the rows exist

- **Scenario:** You are using Oracle databases in your message flows and ESQL binds against columns, which are declared as data type CHAR, and those parameter markers are referenced in a WHERE clause. The database operation fails to return any rows, even though the rows exist.

- **Explanation:** Such fixed length character strings need to be padded with blank characters on Oracle for this type of comparison to succeed.
- **Solution:** Define the CHAR columns as VARCHAR2 columns, or pad the ESQL variable with blank characters to the required column length, so that the comparison locates the desired rows from the table.

There is a memory leak from the Oracle Client Interface on HP-UX 11

- **Scenario:** A memory leak has been observed from the Oracle Client Interface (OCI) when using Oracle 9i Release 2 (9.2.0.1) on HP-UX 11.
- **Solution:** This problem has been fixed by Oracle and the fix is available by upgrading to Oracle 9i Release 2 Database Server Patch Set 2 for HP9000 Series HP-UX (64-bit). This has the Oracle patch number of 2761332. To install the above Oracle patch, you might need to upgrade your Oracle installer (OUI) to version 2.2.0.18.0 first. The Oracle patch number for this upgrade is 2878462.

Stored procedure parameter sizes are different in Oracle 9i and Oracle 10g

- **Scenario:** A stored procedure that contains a char(16) returns a parameter size of 2000 instead of 16.
- **Explanation:** Oracle 9i and Oracle 10g return different sizes for parameters in a stored procedure. This is due to differences in the way that Oracle 9 and Oracle 10 function. A statement from Oracle about this change of behavior, and a scenario that illustrates it, are available at the Oracle meta link web site in TAR 4567976.993.

Error message BIP2731 is issued when you use retained publications with a Sybase database

- **Scenario:** The following error messages are issued when you use retained publications and a Sybase broker database:

  ```
  BIP2731 Database statement 'INSERT INTO dbo.BRETAINEDPUBS
  VALUES(?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)' could not be executed.
  BIP2321 Database error: ODBC return code '-1'.
  BIP2322 Database error: SQL State '40001'; Native Error Code '1205'.
  Text '[SYBASE][ODBC Sybase Driver][SQL Server]Your server command
  (family id #0, process id #234) encountered a deadlock situation.
  Please re-run your command.'
  ```
- **Explanation:** These errors are likely to occur only when you use retained publications with multiple topics, with a substantial workload.
- **Solution:** You must apply row level locking to one of the broker's database tables:
  1. At a Sybase command prompt, enter the command:

     ```
     isql -Uusername -Ppassword
     ```
  2. Connect to the broker's database:

     ```
     use broker DSN
     ```

     where *broker DSN* is the ODBC Data Source Name (DSN) for the broker database.
  3. Enter the command:

     ```
     alter table dbo.BRETAINEDPUBS lock datarows
     ```

     where *dbo* is the schema name.
  4. Enter the command:

     ```
     go
     ```

Database updates are not committed as expected

- **Scenario:** You have included a Compute, Database, or Filter node in a message flow, and you have set the **Transaction** property to `Commit`. The message flow has raised an exception and has rolled back, but the database updates have not been committed.
- **Explanation:** When you set **Transaction** to `Commit`, the database updates performed by the node are committed when the node completes successfully. If an exception is raised before the node has completed, and this causes the message flow to be rolled back, the commit is not issued and the database updates are also rolled back. The situations in which this can occur are:
  - The node itself causes an exception to be raised. The commit is never performed.
  - The ESQL contains a PROPAGATE statement. This statement does not complete until all processing along the path taken by the propagated message has completed, and control returns to the node. Only then can the commit be performed. If an exception is raised along this path, control is not returned and the database updates are rolled back as part of the message flow.
- **Solution:** Review the operation of the node that performs the database updates. For example, it might be possible to split the work between two nodes, with the first updating the database, and the second propagating the output message. Consider changing the ESQL code to process the message in a different way.

The DataDirect trace system is unable to open the ODBC trace file

- **Scenario:** Commands, such as **mqsichangetrace** and **mqsicreatebroker**, fail and a message is displayed stating `Unable to open ODBC trace file`.
- **Explanation:** The DataDirect ODBC file is inaccessible or its size has exceeded 2 GB and tracing has been turned on.
- **Solution:** To resolve this problem, take the following steps:
  1. Turn off tracing in the file to which the ODBCINI environment variable points.
  2. Locate the ODBC trace file to which 'TraceFile' in the ODBC stanza points. If you need to keep the trace, move the ODBC trace file to another location. If you do not need to keep the trace, delete the ODBC trace file.
  3. If you still need to gather ODBC trace, reset `TRACE=1` in the odbc.ini file.

You want to list the database connections that the broker holds

- **Scenario:** You want to list the database connections that the broker holds.
- **Solution:** The broker does not have any functionality to list the connections that it has to a database. Use the facilities that your database supplies to list connections. Refer to the documentation for your database to find out about these.

Resolving problems when using publish/subscribe

Publish/subscribe within WebSphere Message Broker is managed from the workbench, which is used to connect brokers in a topology or hierarchy, and to administer topic-based security (if a User Name Server is in use). Like other workbench changes, you need to deploy any changes that you make so that updates are sent to the broker.

This topic contains advice for dealing with some common problems that can arise when using publish/subscribe:

- "Your broker is not registering with the User Name Server"
- "Application responses are not received"
- "Your application is not receiving publications" on page 91
- "Access control appears not to work" on page 92
- "You cannot access the broker topology after you change the access control list (ACL)" on page 92
- "Publishing a message causes a filter error" on page 92
- "Symbols in subscription filters cause problems" on page 93
- "There are problems with multiple brokers and communications" on page 93
- "The workbench does not display non-Windows users" on page 93
- "There are performance problems on AIX when the JIT compiler is not loaded" on page 94
- "Error message BIP2332E is issued with SQLSTATE 57011 when you access the publish/subscribe database on z/OS" on page 94
- "Your broker does not have the correct authority to run PGM/IP" on page 95
- "Your client does not have the correct authority to run PGM/IP" on page 95
- "Your broker is not running as root" on page 95
- "Your client does not have sufficient memory to run PGM/IP" on page 95

Your broker is not registering with the User Name Server

- **Scenario:** Your broker has been enabled to use topic-based security by setting the **-s** field in the **mqsicreatebroker** command to the User Name Server queue manager, but many occurrences of message BIP8303W are displayed on the console, indicating that the broker's user or group cache is empty. When you try to deploy ACLs, more occurrences of BIP8303W are displayed, as well as the message BIP7017:
- **Explanation:** It is normal to see a few occurrences of BIP8303W, but if you continue to see this message after you deploy ACLs and try to use topic-based security, it is likely that your broker has not been registered with the User Name Server.
- **Solution:**
  - Check that the broker has the correct name for the User Name Server queue manager by going to the broker's file system and looking at the value of the `UserNameServerQueueManagerName` field in the registry entry for the broker. If the value is not correct, use the **mqsichangebroker** command, setting the **-s** flag to the correct User Name Server queue manager.
  - Check that the User Name Server process and the control process are both running. WebSphere MQ intercommunication must also be running correctly if the User Name Server is on a different queue manager.

  In each of these cases, restart the broker, deploy any topics data, and look for the message:

  ```
  +() 0 BIP8204I: User Name Server is registering a client with UUID
  15db2a8e-869e-11d5-8000-091465ac0000, and cache version 0.
  ```

Application responses are not received

- **Scenario:** Application responses are not received.
- **Explanation:** Depending on the application code, a publisher or subscriber might request confirmation that its message was processed successfully. Responses can make debugging client problems much easier, because a response code is given in the event of a problem occurring.

This should always be possible for a subscriber, but for the publishing side, a message might be published without the knowledge of the originating application (for example, by using the default topic property on the input node, or by using a Compute node to modify this property in a message flow). The results of processing that message are still logged in user trace, which might give clues as to what is happening.

If a response is not received, there are several possible causes:

- The system is busy. Messages might build up on the input queue, and the client might not be waiting long enough for its response.
- WebSphere MQ expiry might be being used. There are cases where this is what is required, but the expiry of the input message is copied to the response. As a result, the message might expire on the broker's input queue, or it might expire on the way back to the client. This is not an error, even with a persistent message.
- The input message or response might have been put to the dead-letter queue, if one is configured. Look on this queue to see if any new messages are there. This is normally accompanied by error messages in the broker's log that describe the problem. For example, the reply-to queue might have been incorrectly specified in the input message, so the reply message has been put to the dead-letter queue.

- **Solution:** If your application is not asking for responses (that is, not using messages of type MQMT_REQUEST) consider doing so, particularly when developing applications.

Your application is not receiving publications

- **Scenario:** Your application is not receiving publications.
- **Explanation:** If an application has subscribed successfully (that is, has received an OK response to a RegSub message), it should receive publications that match its subscription. Details of its registration can be viewed in the Subscriptions pane of theBroker Administration perspective. This can be useful to check that the details are correct.

  Subscribers are sent messages only if they match the topic, the subscription point, and the filter. Because the subscription point is specified in the message flow, not in the publication message, an incorrect message flow setting can cause unexpected failures.

  A user trace of the flow containing the Publication node shows you whether publications are matching anything.
- **Solution:** If a filter is being used, a user trace shows you whether this is being evaluated as expected.

  The case with multiple execution groups, or multiple brokers, is more complex. A response is sent to a subscriber once the message has been processed by the target execution group. Other execution groups (and brokers) are updated asynchronously. As a result, there might be a delay before publications made elsewhere are received. If the broker is busy, there can be a delay before messages are processed fully. In a multi-broker setup, if communications have been suspended, subscription changes are propagated through the network of brokers. Check the channels.

  With multiple execution groups or brokers, it might be possible to fill intermediate WebSphere MQ queues if the load is very high. This might be reported in the syslog (if a broker cannot put to a queue because it is full) or in the WebSphere MQ log (if a message coming across a channel cannot be put to the target queue because it is full). If you see messages of this type, display the queue depths on all your queue managers to see if any are almost full.

Access control appears not to work

- **Scenario:** Access control appears not to work.
- **Solution:** Ensure that every broker is configured with the User Name Server. Use a single User Name Server for the whole broker domain. In addition, all the brokers and the Configuration Manager must be able to access the User Name Server's queue manager.

  If BIP8303 errors are recorded in any broker's log after startup, it indicates a communication problem with the User Name Server. Check your WebSphere MQ channels to and from the User Name Server and try again. You should see event message BIP8204I for both the Configuration Manager and any brokers, showing that they have successfully registered with the User Name Server.

  There might be problems with users and group memberships if your broker domain spans a number of systems with different lists of users. For each computer holding a broker, there must be a group called **mqbrkrs**. This must contain the following principals, or authorization failures are reported:

  - Local broker's service user ID
  - All neighboring brokers' service user IDs
  - The Configuration Manager's service user ID

  For client applications, if the user ID in the message is a member of the **mqbrkrs** group, security is bypassed (even if a User Name Server is being used).

  If your publish/subscribe request fails, and event BIP7017 is issued, check that the client user ID is known to the system on which the User Name Server is running. Also, if you are operating in a Windows domain environment, ensure that the User Name Server was created with the **-d** flag on the **mqsicreateusernameserver** command set to the appropriate domain, and that all client application user IDs are members of this domain.

  Message persistence in publish/subscribe is normally preserved. However, a subscriber might not get the expected persistence if ACLs do not allow it.

You cannot access the broker topology after you change the access control list (ACL)

- **Scenario:** After connecting to the Configuration Manager, the broker topology is displayed in the Domains Navigator with restricted access, because your user does not belong to suitable Windows security groups, or an ACL entry has not been defined for this user to access the broker topology. After you have created a suitable ACL entry using the **mqsicreateaclentry** command, and disconnected from, and reconnected to, the domain, the broker topology is still shown with restricted access.
- **Solution:** The ACL information has not yet been refreshed. To force this refresh, stop and restart the User Name Server and the Configuration Manager.

Publishing a message causes a filter error

- **Scenario:** When you publish a message, you receive an error response message with reason text MQRCCF_FILTER_ERROR.
- **Explanation:** A broker returns this message to a publication when subscriptions have been registered that specify filter expressions (for Content Based Routing) and an error has been encountered when the broker attempts to filter the published message. This can occur, for example, if a message is published that includes unsupported data types, or if the message body is corrupted.

Symbols in subscription filters cause problems

- **Scenario:** If you specify certain symbols when you use filters in a subscription, the filter does not work. Sometimes your subscription messages are put to the dead-letter queue, and a number of error messages are written to the local error log indicating MQRFH2 parsing errors.
- **Explanation:** The MQRFH2 header employs standard XML encoding, so its parser interprets some symbols in a special way.
- **Solution:** If you want to include these in your filters, you must use the appropriate escape character to ensure that they are parsed correctly:

Symbol	Escape character
<	&lt;
>	&gt;
″	&quot;
′	&apos;
&	&amp;

For example, if you want to use:

```
<Filter>Body.e_ALERT_BODY.eqnum<6</Filter>
```

you must specify:

```
<Filter>Body.e_ALERT_BODY.eqnum&lt;6</Filter>
```

There are problems with multiple brokers and communications

- **Scenario:** You are experiencing problems with multiple brokers and communications.
- **Explanation:** Channels and transmission queues are needed between each connected broker in the topology. These are configured in the same way as for communication between a single broker and the Configuration Manager. Within a collective, every broker is connected to every other broker, so many channels are needed.
- **Solution:** Using WebSphere MQ clustering might make the administration job easier. Otherwise, ensure that every channel is working.

  When the configuration is deployed, a response is received from each broker; these appear in the workbench. However, you should also check each broker's local log. For each connection, a BIP7113 message is issued. To test communications, the Soccer sample application might prove useful. It is possible to run each part on a separate broker to test communications between them.

The workbench does not display non-Windows users

- **Scenario:** The workbench and Configuration Manager are running on Windows, and the User Name Server and broker are running on z/OS or a Linux or UNIX system. After starting the workbench, you find that the expected list of users and groups from z/OS, Linux or UNIX is not displayed in the Topics view of the Broker Administration perspective.
- **Explanation:** The Configuration Manager has not registered with the User Name Server.
- **Solution:**
  1. Stop the Configuration Manager, using the **mqsistop** command.
  2. Issue the **mqsichangeconfigmgr** command, setting the **-s** flag to the correct User Name Server queue manager name.
  3. Restart the Configuration Manager using the **mqsistart** command.

4. Check that the User Name Server is running.

You need to have both sender and receiver channels and transmission queues established to and from both the Configuration Manager queue manager and the User Name Server queue manager. Check that these channels have been created and are running.

If you make any changes, restart the Configuration Manager and check for Windows event message 8258, which shows that the Configuration Manager has successfully registered with the User Name Server. If you make any changes, for example to channels, you also need to restart the workbench to refresh the displayed list of users.

On z/OS, if the User Name Server is running, and if you have the above options are set correctly, but you are still not seeing z/OS users listed in the Topics view of the Broker Administration perspective, check that the z/OS users all have OMVS segments defined. If they do not, they are not listed by the User Name Server.

There are performance problems on AIX when the JIT compiler is not loaded

- **Scenario:** There are performance problems on AIX when the JIT compiler is not loaded.
- **Explanation:** The environment variable LIBPATH can affect the loading of the Java JIT (just-in-time) compiler on AIX. WebSphere Message Broker for AIX will run correctly without the JIT compiler, but publish/subscribe performance will be adversely impacted. You are therefore recommended to ensure that the JIT compiler runs.

  The JIT compiler will load and run correctly if LIBPATH is not set. This is the simpler case and you are recommended to not set LIBPATH. You can make libraries available by linking them into /var/wmqi/lib (for all WebSphere Message Broker for AIX processes) or /usr/lib (for all processes on the system). WebSphere Message Broker for AIX configuration does this for DB2 libraries.

  If it is necessary to set LIBPATH, update it to include the directory /usr/java130/bin. For example, you can use the following command to start the broker:

  ```
  LIBPATH=/usr/local/lib:/usr/java130/bin mqsistart mybroker
  ```

Error message BIP2332E is issued with SQLSTATE 57011 when you access the publish/subscribe database on z/OS

- **Scenario:** You are using publish/subscribe on z/OS with the MQ transport and large numbers of retained publications or subscriptions are being changed. Error message BIP2332 is issued with SQLSTATE 57011 when you access the publish/subscribe database. You might also see DB2 errors in your SDF log; for example:

  ```
  #DFL0 DSNILMCL RESOURCE UNAVAILABLE 558
    CORRELATION-ID=ST03BRK
    CONNECTION-ID=RRSAF
    LUW-ID=*
    REASON 00C90092
    TYPE 00000905
    NAME IRLM
  *DXR175E IFL0001 IRLM IS UNABLE TO OBTAIN STORAGE - ECSA
   DSNT501I  #DFL0 DSNILMCL RESOURCE UNAVAILABLE 560
    CORRELATION-ID=ST04BRK
    CONNECTION-ID=RRSAF
  ```

```
LUW-ID=*
REASON 00C90092
TYPE 00000905
NAME IRLM
```

- **Explanation:** This is a DB2 tuning issue with the IRLM (lock manager) subcomponent.
- **Solution:** Make sure that you run IRLM with the lock information being held in local storage; use the startup parameter **PC=YES**. There is more information about this in the DB2 Redbook *DB2 UDB for OS/390 Version 6 Performance Topics, SG24-5351.*

Your broker does not have the correct authority to run PGM/IP

- **Scenario:** The broker does not have the correct authority to run and has reverted back to PGM/UDP.
- **Explanation:** The broker is configured to use PGM/IP but does not have the correct user authority. The broker has reverted back to PGM/UDP.
- **Solution:** Specify a valid user authority for the broker, or select a different protocol using the multicast properties panel or the mqsichangeproperties command.

Your client does not have the correct authority to run PGM/IP

- **Scenario:** An exception occurred when the client attempted to connect with the broker.
- **Explanation:** The client is configured to use PGM/IP but does not have the correct user authority. An exception occurred when the client tried to connect with the broker.
- **Solution:** Obtain the correct user authority for your client.

Your broker is not running as root

- **Scenario:**The broker is not running as root.
- **Explanation:** The broker was configured to use PGM/IP but was not running as root. It has reverted back to PGM/UDP.
- **Solution:** Obtain root access and ensure the broker is running as root (as Administrator or as a regular user who is a member of the Administrators group on Windows), in a secure environment. Be aware of the security implications of doing this.

Your client does not have sufficient memory to run PGM/IP

- **Scenario:** Your client does not have sufficient memory to run PGM/IP
- **Explanation:** Clients running PGM/IP require the multicast property Maximum Memory Size to be set to a suitable value.
- **Solution:** Change the value of Maximum Client Memory Size parameter using the multicast properties panel or the mqsichangeproperties command. The Maximum Client Memory Size parameter is only applicable to PGM protocols, and is greyed out unless Protocol Type is set to one of the PGM protocols. This parameter is available in the Advanced properties panel.

Resolving problems with performance

- **Scenario:** You are experiencing problems with performance, such as:
  - Poor response times in the workbench when developing message flows
  - Poor response time at deployment
  - Individual messages taking a long time to process

  - Poor overall performance, or performance that does not scale well
- **Solution:** Possible solutions are:
  - Speed up WebSphere MQ persistent messaging by optimizing the I/O (Input/Output)
  - Speed up database access by optimizing I/O
  - Increase system memory
  - Use additional instances or multiple execution groups
  - Optimize ESQL statements for best performance

A good source of information about performance is the set of reports in WebSphere MQ Family Category 2 (freeware) SupportPacs, available for download from the WebSphere MQ SupportPacs Web page.

This topic contains advice for dealing with some common performance problems that can arise:

- "Your system is getting increasingly slower"
- "You experience configuration problems with a large number of components"
- "A WHILE loop in a large XML array takes a long time to process" on page 97
- "Remote waitForMessages calls with WebSphere MQ Everyplace are slow" on page 98
- "There is a loss in performance for stored procedures" on page 98

Your system is getting increasingly slower

- **Scenario:** The system is getting increasingly slower.
- **Explanation:** When a process ends abnormally, an entry is made in the local error log, and a dump data file might be written to the errors or z/OS log directory. This directory is unbounded, and if you do not clear it of unwanted files you might find that your system performance degrades due to significant space being used up by old abend files.
- **Solution:** On Windows, you are advised to use the **-w** flag of the **mqsicreatebroker** command to create the errors directory in a hard drive partition that does not contain WebSphere Message Broker or Windows itself.

You experience configuration problems with a large number of components

- **Scenario:** You are experiencing configuration problems with a large number of components.
- **Explanation:** If you are running WebSphere Message Broker with a large number of configured components, the memory footprint of the broker processes (particularly the DataFlowEngine) might exceed their memory limits. In particular, the user process limit might be exceeded, or the address space limit might be reached. You might encounter problems, such as the BIP2106E error message, when running a broker with:
  - A large number of message flows
  - Multiple databases
  - Very large input or output messages
- **Solution:** Use tools that are specific to your operating system to check the maximum size of the failing process, then check for any user limits (if applicable) or computer limits on process size.
  1. Use the **ulimit** command to check user limits on Linux or UNIX systems and increased if needed.

There is a hard limit on each operating system for the maximum process size, beyond which failure is inevitable:

Operating system	Maximum process size
Windows	2 GB
Solaris	About 3.75 GB
HP-UX	About 3 GB, depending on kernel settings
AIX (32 bit system)	2 GB
z/OS	2 GB

Increasing user limits beyond these values will not make any difference. If your broker processes regularly reach these sizes, consider spreading your message flows across more execution groups to reduce the size of each one below these limits.

2. On AIX, WebSphere Message Broker limits DataFlowEngines to two 256 MB memory segments, that is, 512 MB. Normally DataFlowEngine processes do not require more than 512 MB of the AIX address space, so the DataFlowEngine executable file is linked so as to have a default address space of two segments. However, it is possible to extend the size of the AIX address space available to the DataFlowEngine process by increasing the number of 256 MB memory segments available to it. You can patch the DataFlowEngine to use more memory segments using the following shell command:

```
/usr/bin/echo '\0100\0\0\0'|dd of=executable_file_name bs=4 count=1 seek=19 conv=notrunc
```

This creates a version of the DataFlowEngine that uses four segments (that is, 1 GB).

Alternatively, on AIX 5.1 and above, you can achieve the same result by using the **ldedit**command:

```
ldedit -b maxdata:0x40000000 DataFlowEngine
```

The application of fix pack maintenance replaces the existing DataFlowEngine, so if you have used the above process, you need to repeat it after every fix pack installation.

The technique described above overrides only the soft limit and not the hard limit. If your broker processes regularly reach these sizes, consider spreading your message flows across more execution groups to reduce the size of each one to be below these limits.

A WHILE loop in a large XML array takes a long time to process

- **Scenario:** A WHILE loop in a large XML array takes a long time to process.
- **Explanation:** You are probably using the CARDINALITY() function to determine the size of the array in the WHILE statement. This is not recommended, because with large arrays the cost of determining the number of elements is proportional to the size of the array. The CARDINALITY function is invoked each time the WHILE statement is executed. The message has many elements, so there is a significant overhead when running the loop in this way.
- **Solution:** Unless you have a message in which the number of elements of the array grows dynamically, determine the size of the array before entering the WHILE loop. Use code similar to the following:

```
DECLARE i,c INTEGER;
SET i = 1;
SET c=CARDINALITY(OutputRoot.XML.MyMessage.Array[ ]);
WHILE i <= c DO
      . . .
      . . .  loop processing
      . . .
END WHILE;
```

Remote waitForMessages calls with WebSphere MQ Everyplace are slow

- **Scenario:** Remote waitForMessages calls with WebSphere MQ Everyplace are slow.
- **Explanation:** This is because a remote waitForMessages() call necessarily results in a polling action: the client queue manager attempts to get a message repeatedly until one is available on the remote queue or a timeout is reached.
- **Solution:** Define a store-and-forward queue on the broker's queue manager and a home-server queue on the client. This provides a level of de-coupling that allows the arrangement to work in the event of the client queue manager becoming unavailable. Alternatively, specify a remote queue in the WebSphere MQ Everyplace output node, so that a local GET call is sufficient on the client side.

There is a loss in performance for stored procedures

- **Scenario:** Stored procedures that do not return results sets do not perform as well as they did when using the DataDirect version 4.1 drivers.
- **Explanation:** The new DataDirect version 5 Oracle drivers now use the configuration parameter **ProcedureRetResults** to allow the driver to return results sets from stored procedures. This parameter applies only to Oracle. By default, the parameter is set to 1 and has to be 1 if you use stored procedures that return results sets. When this parameter is set to 0, the driver does not return results sets from stored procedures and this can result in better performance.
- **Solution:** If your stored procedures do not return any results sets, set the **ProcedureRetResults** parameter to 0 (zero).

Resolving problems when developing Configuration Manager Proxy applications

This topic contains advice for dealing with some common problems that can arise when developing Configuration Manager Proxy (CMP) applications:

- "Your Configuration Manager Proxy application hangs if the Configuration Manager is not available"
- "You set a property of an object and query its value, but the value has not changed" on page 99
- "You cannot connect to a Configuration Manager using .CONFIGMGR files" on page 99
- "You tried to move a broker between Configuration Managers and now you cannot see the broker's resources" on page 99

Your Configuration Manager Proxy application hangs if the Configuration Manager is not available

- **Scenario:** When the Configuration Manager is unavailable, your Configuration Manager Proxy application hangs.

- **Explanation:** Communication between the Configuration Manager Proxy and the Configuration Manager is asynchronous, so the Configuration Manager Proxy hangs because it is waiting for a message from the Configuration Manager.
- **Solution:** Configure the maximum amount of time that the Configuration Manager Proxy waits by using the following method:

  ```
  // Wait for a maximum of 10 seconds
  ConfigManagerProxy.setRetryCharacteristics(10000);
  ```

  This represents the time in milliseconds that the Configuration Manager Proxy will wait for information before throwing the *ConfigManagerProxyPropertyNotInitializedException* exception.

  If you set this timeout value too low, an exception will be thrown even if the Configuration Manager is available.

You set a property of an object and query its value, but the value has not changed

- **Scenario:** You have set a property of an object, then queried its value; the value has not changed.
- **Explanation:** Methods that change properties of domain objects are not processed immediately. If you call a property change method on a Configuration Manager Proxy object, this causes the Configuration Manager Proxy to send a message to the Configuration Manager asking it to make the requested change. The Configuration Manager processes the request asynchronously and notifies any *AdministeredObjectListeners* of the affected object when the change has been attempted.
- **Solution:** Methods that change state usually return as soon as the request has been put to the Configuration Manager's queue manager, or following a call to *ConfigManagerProxy.beginUpdates()*, as soon as the request has been added to the current batch. If the property has still not been updated after the action's response to the request has been returned to the Configuration Manager Proxy, consult the response message for more details.

You cannot connect to a Configuration Manager using .CONFIGMGR files

- **Scenario:** You cannot connect to a Configuration Manager using .CONFIGMGR files.
- **Explanation:** The *MQPropertyFileConfigManagerConnectionParameters* class allows Configuration Manager Proxy applications to connect to Configuration Managers using files with a .configmgr extension, although it requires an XML parser in order to do this.
- **Solution:** Ensure that a supported parser is available on the CLASSPATH. A supported parser is shipped with the product.

  Alternatively, use the *MQConfigManagerConnectionParameters* class instead of the *MQPropertyFileConfigManagerConnectionParameters* class. This allows you to connect to Configuration Managers by specifying the host name, queue manager name and port of the target Configuration Manager directly. This method does not require an XML parser.

You tried to move a broker between Configuration Managers and now you cannot see the broker's resources

- **Scenario:** You used the BrokerProxy.setUUID() method to attempt to move a broker from one Configuration Manager to another and you can no longer see the broker's execution groups, message flows, message sets, and subscriptions.
- **Explanation:** This is a limitation of the setUUID() method.

- **Solution:** Remove all deployed resources and redeploy them to the broker.

Resolving problems with user-defined extensions

This topic contains advice for dealing with some common problems that can arise when working with user-defined extensions:

- "You cannot deploy one of your user-defined nodes, despite having a plug-in LIL in the correct directory."
- "You get problems when nodes try to use the ESQL path interface in the plug-in API"
- "Interpreting problems in user-defined extensions" on page 101
- "You are migrating a C user-defined node and cniDefineNodeClass returns CCI_INV_IMPL_FUNCTION." on page 104
- "You see an error when deploying a user-defined extension on z/OS" on page 103
- "You cannot determine which user-defined extensions have been loaded by the broker on startup" on page 103
- "You want to debug classloading" on page 103

You cannot deploy one of your user-defined nodes, despite having a plug-in LIL in the correct directory.

- **Scenario:** You cannot deploy one of your user-defined nodes, despite having a plug-in LIL in the correct directory.
- **Explanation:** You have `memset()` the data area to zero and have not initialized the CNI_VFT structure with the initialization constant `{CNI_VFT_DEFAULT}`.
- **Solution:** Initialize by copying a predefined initialization structure over the function table area, as follows:

  ```
  static CNI_VFT virtualFunctionTable = {CNI_VFT_DEFAULT};
  ```

  In addition, implement logging from your user-defined node so that you can see if the plug-in API is producing error codes; the broker does not log these to its own log, unless you take a service trace.

You get problems when nodes try to use the ESQL path interface in the plug-in API

- **Scenario:** When you attempt to use the ESQL path interface in the plug-in API, the return value is CCI_PATH_NOT_NAVIGABLE.
- **Explanation:** The plug-in API allows you to specify a path in the form of an ESQL path expression and navigate to that element, returning a handle to it if it exists. It also allows you to create elements along the path to the requested element.

  The navigate path utility function (`cniSqlNavigatePath`) executes the SQLPathExpression created with the `cniSqlCreateReadOnlyPathExpression` or `cniSqlCreateModifiablePathExpression` utility functions, as defined by the `sqlPathExpression` argument.

  If the path is not navigable, the return code is set to CCI_PATH_NOT_NAVIGABLE. This might be returned when embedding a path expression in another path expression. The input `cciMessage*` functions must not be NULL; however, any of the output `cciMessage*` functions can be NULL. If you embed a path expression that can be NULL inside a path expression that cannot be NULL, CCI_PATH_NOT_NAVIGABLE is returned.

- **Solution:** If the return code is set to CCI_PATH_NOT_NAVIGABLE, ensure that if a correlation name is specified in a path, the respective parameter is not NULL.

Interpreting problems in user-defined extensions

- **Scenario:** You want to debug problems in user-defined nodes and parsers.
- **Solution:** Start user trace at debug level. In order to see BIP4142, BIP4144, BIP4145, and BIP4146, this must be done at the execution group level. For example, use the mqsichangetrace command without the `-f` flag.

  The following debug messages are available to help you to understand the execution of your user-defined nodes and parsers:

  – BIP2233 and BIP2234: a pair of messages traced before and after a user-defined extension implementation function is invoked. These messages report the input parameters and returned value. For example:

    BIP2233 `Invoking user-defined extension function [function name] ([function call parameters])`

    BIP2234 `Returned from user-defined extension function [function name] with result: [result of call]`

    Note: In these messages, an ″implementation function″ can be interpreted as either a C implementation function or a Java implementation method.

  – BIP3904: a message traced before invoking the Java `evaluate()` method of a user-defined node. For example:

    BIP3904 `(for Java): Invoking the evaluate() method of node (class=[node class name], name=[label of node in flow])` where ″node class name″ is the name of the Java user-defined extension class.

  – BIP3905: a message traced before invoking the C `cniEvaluate` implementation function (iFpEvaluate member of CNI_VFT) of a user-defined node. For example:

    BIP3905 `(for C): Invoking the cniEvaluate() implementation function of node (class=[node class name], name=[label of node in flow])` where ″node class name″ is the name of the user-defined extension class that is provided by the user-defined extension while calling C cniDefineNodeClass.

  – BIP4142: a debug message that is traced when invoking a user-defined node utility function, where the utility function alters the state of a syntax element. This includes all utility functions that start with `cniSetElement`*, where * represents all nodes with that stem.

    BIP4142 `Evaluating cniSetElement [element identifier type]. Changing value from [value before user’s change] to [value after user’s change]″`

  – BIP4144 and BIP4145: a pair of messages traced by certain implementation functions that, when invoked by a user-defined extension, can modify the internal state of a message broker’s object. Possible message broker objects include syntax element, node, and parser. These messages report the input parameter provided to the invoked method and the returned value. For example:

    BIP4144 `Entered function [function name] ([function call parameters])`

    BIP4145 `Exiting function [function name] with result: [result to be returned]`

    In these messages, an ″implementation function″ can be interpreted as either a C implementation function or a Java implementation method.

    The C implementation functions that invoke messages BIP4144 and BIP4145 include:

For user-defined parsers	For user-defined nodes
cpiCreateParserFactory	cniCreateElement*
cpiDefineParserClass	cniDeleteMessage
cpiAppendToBuffer	cniAdd*
cpiCreateElement	cniDetach
cpiCreateAndInitializeElement	cniCopyElementTree
cpiAddBefore	cniFinalize
cpiAddAfter	cniWriteBuffer
cpiAddAsFirstChild	cniSql*
cpiAddAsLastChild	cniSetInputBuffer
cpiSetNameFromBuffer	cniDispatchThread

(* represents all nodes with that stem; for example, cniAdd* includes cniAddAfter, cniAddasFirstChild, cniAddasLastChild, and cniAddBefore.)

The Java methods that invoke messages BIP4144 and BIP4145 are:

For user-defined nodes
com.ibm.broker.plugin.MbElement.CreateElement*
com.ibm.broker.plugin.MbElement.add*
com.ibm.broker.plugin.MbElement.detach
com.ibm.broker.plugin.MbElement.copyElementTree

– BIP4146: a debug message that is traced when invoking a user-defined parser utility function, where the utility function alters the state of a syntax element. This includes all utility functions that start with `cpiSetElement*`, where * represents all nodes with that stem. For example:

```
BIP4146 Evaluating cpiSetElement [element identifier type]. Changing
value from [value before user’s change] to [value after user’s change]
```

For information on the C user-defined API, see the C user-defined parser API and the C user-defined node API.

– BIP4147: an error message that is traced when a user-defined extension passes an invalid input object to a user-defined extension utility API function. For Example:

```
BIP4147 User-defined extension input parameter failed debug validation
check. Input parameter [parameter name] passed into function [function
name] is not a valid object.
```

– BIP4148: an error message that is traced when a user-defined extension damages a broker’s object. For Example:

```
BIP4148 User-defined extension damaged broker's object. Function
[function name] has damaged broker's object passed as parameter
[parameter name].
```

– BIP4149: an error message that is traced when a user-defined extension passes an invalid input data pointer to a user-defined extension utility API function. For Example:

```
BIP4149 User-defined extension input parameter failed debug validation
check. Input parameter [parameter name] passed into function [function
name] is a NULL pointer.
```

  - BIP4150: an error message that is traced out when a user-defined extension passes invalid input data to a user-defined extension utility API function. For example:

    ```
    BIP4150 User-defined extension input parameter failed debug validation
    check. Input parameter [parameter name] passed into function [function
    name] does not have a valid value.
    ```

You want to debug classloading

- **Scenario:** You want to debug classloading.
- **Solution:** Classes and the location from which they are loaded are written to user trace. Use this information to check that the correct classes are being loaded.

You see an error when deploying a user-defined extension on z/OS

- **Scenario:** When you deploy a user-defined extension on z/OS, Linux, or UNIX, you see an error in the log of each execution group stating that there is insufficient authority to open the lil file.
- **Explanation:** On Linux and UNIX, the user-defined extension must have group read permission. On z/OS, the user-defined extension must have group execute permission.
- **Solution:**
  - On Linux and UNIX, set the file permissions of the user-defined extension to group read by issuing the command **chmod a+r**.
  - On z/OS, set the file permissions of the user-defined extension to group read and execute by issuing the command **chmod a+rx**.

You cannot determine which user-defined extensions have been loaded by the broker on startup

- **Scenario:** You cannot determine which user-defined extensions have been loaded by the broker on startup.
- **Solution:** Use the **mqsireportproperties** command for each type of user-defined extension.
  - For a Java user-defined extension, issue the command:

    ```
    mqsireportproperties WBRK_BROKER -e default -o ComIbmJavaPluginNodeFactory -r
    ```

    You see a report similar to this example:

    ```
    ComIbmJavaPluginNodeFactory
     uuid='ComIbmJavaPluginNodeFactory'
     userTraceLevel='none'
     traceLevel='none'
     userTraceFilter='none'
     traceFilter='none'
     NodeClassName='ComIbmJMSClientInputNode'
     NodeClassName='ComIbmJMSClientOutputNode'
     NodeClassName='ComIbmJavaComputeNode'
     NodeClassName='ComIbmXslMqsiNode'
     NodeClassName='SearchFilterNode'

    BIP8071I: Successful command completion.
    ```

    The user-defined extension called SearchFilter has a NodeClassName of SearchFilterNode.
  - For a C user-defined extension (assuming that CONST_PLUGIN_NODE_FACTORY was set to ComIbmSamplePluginNodeFactory in the NodeFactory.h file, as in the sample NumComputeNode), issue the command:

```
mqsireportproperties WBRK_BROKER -e default -o ComIbmSamplePluginNodeFactory -r
```

You see a report similar to this example:

```
ComIbmSamplePluginNodeFactory
 uuid='ComIbmSamplePluginNodeFactory'
 userTraceLevel='none'
 traceLevel='none'
 userTraceFilter='none'
 traceFilter='none'
 NodeClassName='NumComputeNode'

BIP8071I: Successful command completion.
```

The user-defined extension called NumCompute has a NodeClassName of NumComputeNode.

You are migrating a C user-defined node and cniDefineNodeClass returns CCI_INV_IMPL_FUNCTION.

- **Scenario:** When you attempt to migrate a C user-defined node, `cniDefineNodeClass` returns CCI_INV_IMPL_FUNCTION.
- **Explanation:** New fields have been added to the CNI_VFT struct. CNI_VFT_DEFAULT has been updated to initialise these new fields in the V6 header file BipCci.h. If you initialise your CNI_VFT with CNI_VFT_DEFAULT, you should not need to make any code changes. However, if you do not initialise CNI_VFT with CNI_VFT_DEFAULT, these new fields are initialised with random values.
- **Solution:** Initialise your CNI_VFT with CNI_VFT_DEFAULT.

Using logs

This section describes how to view the various logs available to you with WebSphere Message Broker, and how to interpret the information in those logs. It contains the following topic areas:

Local error log

- "Windows: Viewing the local error log"
- "Linux and UNIX systems: Configuring the syslog daemon" on page 105
- "z/OS: Viewing the broker joblogs" on page 106

Eclipse log

- "Viewing the Eclipse error log" on page 107

WebSphere MQ log

- "Viewing WebSphere MQ log information" on page 108

Database log

- "Viewing database log information" on page 108

Windows: Viewing the local error log

The Windows Event Viewer is where WebSphere Message Broker writes records to the local system.

Viewing the system log

The system log contains events logged by the Windows system components. For example, the failure of a driver or other system component to load during startup is recorded in the system log. To view the system log:

1. Open a command prompt.
2. At the prompt type `eventvwr`. This opens the Windows Event Viewer.
3. In the left pane of this window, click the **System Log**(this is often selected as default). Here all the events that have been written to the local system are displayed.

Viewing the application log

The application log contains events logged by applications or programs. For example, a database program might record a file error in the application log. To view the application log:

1. Open a command prompt.
2. At the prompt type `eventvwr`. This opens the Windows Event Viewer.
3. In the left pane of this window, click the **Application Log**. Here all the events that have been logged by applications or programs are displayed.

Interpreting log information

In both logs, each event is displayed on a separate row, in date and time order (most recent first), with the following information:

- **Type:** The event type, which can be information, a warning, or an error.
- **Date and time:** The date and time when the event was written to the log.
- **Source:** What action has caused the event.
- **Category:** The category of the event. The default category is `none`.
- **Event:** The event number.
- **User:** The name of the user at the time of the event.
- **Computer:** The name of the local machine.

To view an individual log entry:

1. Within the system or application log, find the log entry.
2. Right-click the entry and from the pop-up menu click **Properties**. This opens the Information Properties window, where there is a description of the event. There is also a Data section that details bytes or words that were parsed when the record was written to the log.
3. In the Information Properties window, use the up and down arrows to move through the events of the log.
4. To close the Information Properties window, click **OK**; this returns you to the system or application log.

Linux and UNIX systems: Configuring the syslog daemon

On Linux and UNIX systems, all WebSphere Message Broker messages (other than those generated by the command line utilities) are sent to the syslog, so it is useful to redirect user messages to a separate file.

Before you create a broker on Linux or UNIX systems, configure the syslog daemon to redirect user messages to a file called user.log:

1. Log on as root.
2. Enter the following commands to create a file called user.log:

   On UNIX systems enter the command:

```
touch /var/adm/user.log
chown root:mqbrkrs /var/adm/user.log
chmod 640 /var/adm/user.log
```

On Linux enter the command:

```
touch /var/log/user.log
chown root:mqbrkrs /var/log/user.log
chmod 640 /var/log/user.log
```

3. Add the following line to the /etc/syslog.conf file to redirect debug level messages to the file user.log:

   On UNIX systems enter the command:

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

   On Linux enter the command:

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

   You can use `user.* -` instead of `user info` in the preceding examples.

   - `*` means that information, notice, warning, and debug messages are caught
   - `-` means that `syslog` does not synchronize the file after writing to it.

   There is a performance gain, although you might lose some data if the machine crashes immediately after it has written to the file.You can add similar lines to direct information, warning, and error messages to user.log.
4. Restart the syslog daemon.

   On AIX enter the command:

   ```
   refresh -s syslogd
   ```

   On HP-UX and Solaris, enter the command:

   ```
   kill -HUP 'cat /etc/syslog.pid'
   ```

   On Linux enter the command:

   ```
   /etc/init.d/syslogd restart
   ```

   or

   ```
   /etc/rc.d/init.d/syslogd restart
   ```

   for systems where `rc.d` is not a soft link

For other syslog options, see the documentation for your platform.

z/OS: Viewing the broker joblogs

On z/OS, the broker writes messages to the appropriate z/OS system log and joblogs. These might include information, warning, error, and severe messages to indicate various situations and events.

Understanding the broker address spaces

The broker run time consists of multiple address spaces. There is a single control address space that is always running when the broker is active and is responsible for communicating with the Configuration Manager and workbench. Each execution group within the broker is mapped to its own address space and as these execution groups start and stop, the corresponding address spaces are started and stopped. The control address space is assigned a JOBNAME and STEPNAME, which is the same as the broker name. The execution groups have a JOBNAME that is also the same as the broker name, and a STEPNAME that matches the last seven characters of the execution group name.

Viewing the z/OS system console log

The broker writes all of its messages to the z/OS system console log. You can see messages from all address spaces running on the z/OS system in this log. It is easy to identify jobs associated with the broker in the console log because of the naming of broker address spaces. Using the console log, you can see the order of event reporting for different products. This is helpful for cross-product problem determination.

Viewing the broker joblogs

The broker control address space and each of the execution group address spaces has its own joblog. Selecting the joblog for the appropriate broker address space allows you to see all messages relating to that address space. This is helpful in a busy system where the system console log might have many messages from different products obscuring the information in which you are interested.

Interpreting log information

In both logs, each message is displayed on its own, in date and time order; it might span multiple lines, if necessary. For each message, the following information is available:

- **Date and time**: The exact date and time when the event was written to the log.
- **JOBID**: The JOBID of the address space. This is the started task job identifier.
- **Message Number**: The message number, which identifies whether the event is information, a warning, or an error, with diagnostic text.
- **JOBNAME**: The JOBNAME of the address space issuing the message. This is always the same as the broker name.
- **STEPNAME**: The STEPNAME of the address space issuing the message. For the control address space, this is the same as the broker name; otherwise, it is the same as the last seven characters of the execution group issuing the message.

Viewing the Eclipse error log

The Eclipse error log captures internal errors that are caused by the operating system or your code.

To view the Eclipse error log:

1. Switch to the Plug-in Development perspective.
2. In the Tasks view, in the bottom right pane, click the **Error Log**tab.

   The error log is displayed, showing the following information for each error:
   - The status of the error (for example, error or warning)
   - A brief description of the error
   - Which plug-in the error derived from
   - The date and time that the error was produced
3. If an error has a plus sign (+) at the start of it, this means that it is a complex problem, and there are a number of errors contributing to it. Click the plus sign to view the individual errors.
4. To see the details of a particular problem, double-click the entry in the Tasks view. A separate window is displayed showing more details of the error.

Viewing WebSphere MQ log information

WebSphere MQ messages are written to the local error log in the same way as WebSphere Message Broker messages. WebSphere MQ messages start with the prefix AMQ followed by a number between 4000 and 9999. The number specifies which part of WebSphere MQ the message originates from:

4000 to 4999
Windows user interface messages

5000 to 5999
Installable services messages

6000 to 6999
Common services messages

7000 to 7999
WebSphere MQ product messages

8000 to 8999
WebSphere MQ administration messages

9000 to 9999
Remote messages

For a full explanation of each message, see the WebSphere MQ library Web page.

Viewing database log information

Databases write severe error and warning conditions to the local error log (syslog). Typically, databases also write errors to a database log file, such as the db2diag.log file on DB2.

You can open the db2diag.log file in a text editor.

The example below shows a typical entry from the db2diag.log file:

```
Jun 18 15:02:53 bluj DB2[46827]: DB2(db2inst1.000(1))oper_system_services sqlobeep(2)
 reports:(3)
Jun 18 15:02:53 (4) bluj (5) DB2[46827(6)]: extra symptom string provided:(7) RIDS/sqlesysc_
Jun 18 15:02:53 bluj DB2[46827]:     data: (8)  54686973  20697320  616e2065  78616d70
Jun 18 15:02:53 bluj DB2[46827]:     data:   6c65206f  66206c6f  67676564  20646174
Jun 18 15:02:53 bluj DB2[46827]:     data:   61
Jun 18 15:02:53 bluj DB2[46827]: 2 piece(s) of dump data provided... to file(9) /u/db2inst1/
Jun 18 15:02:53 bluj DB2[46827]:         1. 'DUMP EXAMPLE #1' has been dumped (10)
Jun 18 15:02:53 bluj DB2[46827]:         2. 'DUMP EXAMPLE #2' has been dumped
```

The bold numbers in the example show various items in the log file:

(1) The instance name and node number.

(2) The reporting component and function.

(3) The probe ID and error and alert numbers.

(4) A time stamp for when the event occurred.

(5) The host name.

(6) The process ID of the reporting process. Use the **ps** command to view information about the process ID of the reporting process. For example, enter the following command to get information about the reporting process:

```
ps -fu 46827
```

(7) A symptom string that contains additional information about where and why the problem occurred.

(8) A hexadecimal dump of data that includes return codes and other information that can be interpreted by your IBM Support Center.

(9) Information about additional dump files. Larger structures and other binary data might be dumped to additional files. The name of the file is identified in the syslog file.

(10) An entry to identify a piece of dump data.

Using trace

This section describes how to use the optional trace: user trace, service trace, ODBC trace, WebSphere MQ Java Client trace, and Configuration Manager Proxy trace. It contains the following topics:

For user trace:

- "Starting user trace"
- "Checking user trace options" on page 110
- "Changing user trace options" on page 111
- "Retrieving user trace" on page 112
- "Stopping user trace" on page 111

For service trace:

- "Starting service trace" on page 112
- "Checking service trace options" on page 113
- "Changing service trace options" on page 114
- "Retrieving service trace" on page 114
- "Stopping service trace" on page 114

For both types of trace:

- "Formatting trace" on page 115
- "Interpreting trace" on page 116
- "Clearing old information from trace files" on page 117
- "Changing trace settings from the workbench" on page 117

Other types of trace:

- "ODBC trace" on page 118
- "WebSphere MQ Java Client trace" on page 119
- "Configuration Manager Proxy trace" on page 119

Starting user trace

Before you start:

Before you start to trace a broker, or any of its execution groups or message flows, the broker must be running, and you must have deployed the message flows using the workbench. Follow the instructions in Deploying.

To start a user trace:

1. Start WebSphere Message Broker user trace facilities by using the **mqsichangetrace** command, or, for execution groups and assigned message

flows, from the workbench. You can select only one broker on each invocation of the command, but you can activate concurrent traces for more than one broker, by invoking the command more than once.

2. Specify an individual execution group or message flow within the specified broker to limit the scope of a trace. The events recorded when you select the message flow option include:
   - Sending a message from one Message Processing node to the next.
   - Evaluating expressions in a Filter or Compute node.
3. Start your trace. You can start trace at two levels:

   `normal` This tracks events that affect objects that you create and delete, such as nodes.

   `debug` This tracks the beginning and end of a process, as well as monitoring objects that are affected by that process.

Example: starting user trace for the default execution group

To start normal level user tracing for the default execution group on a broker that you have created with the name `WBRK_BROKER`, on distributed platforms, enter the command `mqsichangetrace WBRK_BROKER -u -e default -l normal` where:

-u specifies user trace
-e specifies the execution group (in this case, the default execution group)
-l specifies the level of trace (in this case, normal)

On z/OS, enter the command `F MQP1BRK,ct u=yes, e='default', l=normal`

Example: starting user trace for the default execution group from the workbench

To start normal level user tracing for the default execution group from the workbench:

1. Switch to the Broker Administration perspective.
2. In the Domains view, right-click the default execution group.
3. Click **User trace** → **Normal**.

Example: starting user trace for a message flow from the workbench

To start normal level user tracing for one of your message flows from the workbench:

1. Switch to the Broker Administration perspective.
2. In the Domains view, right-click the message flow for which you want to start trace.
3. Click **User trace** → **Normal**.

An alert saying `Message Flow is tracing at level 'normal'` is displayed in the Alert Viewer.

Checking user trace options

Use the **mqsireporttrace** command to check what tracing options are currently active for your brokers.

Specify the component for which the check is required, for example, the execution group or message flow. The command responds with the current trace status for the component you have specified.

Example: checking user trace options for a broker

To check what options are currently set for the broker WBRK_BROKER and its execution group test, on distributed platforms, enter the command mqsireporttrace WBRK_BROKER -u -e test where:

-u specifies user trace
-e specifies the execution group (in this case, test)

On z/OS, enter the command F MQP1BRK,reporttrace u=yes, e='test'

If you have started tracing by following the example in "Starting user trace" on page 109, the response to the **mqsireporttrace** command is:

```
BIP8098I: Trace level: normal, mode: safe, size: 1024 KB
BIP8071I: Successful command completion
```

Changing user trace options

Use the **mqsichangetrace** command to change the trace options that you have set. You can also use the workbench to change the trace options for execution groups and assigned message flows.

Example: changing user trace from normal to debug

To change from a normal level of user trace to a debug level on the default execution group of a broker called WBRK_BROKER, on distributed platforms, enter the command mqsichangetrace WBRK_BROKER -u -e default -l debug where:

-u specifies user trace
-e specifies the execution group (in this case, the default execution group)
-l specifies the level of trace (in this case, changing it to debug)

On z/OS, enter the command F MQP1BRK,ct u=yes, e='default', l=debug

Example: changing user trace from normal to debug from the workbench

To change from a normal level of user trace to a debug level for one of your message flows from the workbench:

1. Switch to the Broker Administration perspective.
2. In the Domains view, right-click the message flow for which you want to change trace.
3. Click **User trace** → **Debug**.

An alert saying Message Flow is tracing at level 'debug' is displayed in the Alert Viewer.

Stopping user trace

Use the **mqsichangetrace** command with a trace level of none to stop an active trace. This stops the trace activity for the component that you specify on the command. It does not affect active traces on other components. For example, if you stop tracing on the execution group `test`, an active trace on another execution group continues.

You can also stop an active trace for execution groups or assigned message flows using the workbench. If you redeploy a component from the workbench, trace for that component is returned to its default setting of `none`.

Example: stopping user trace on the default execution group

To stop the trace started by the command shown in "Starting user trace" on page 109, on distributed platforms, enter the command `mqsichangetrace WBRK_BROKER -u -e default -l none` where:

-u specifies user trace
-e specifies the execution group (in this case, the default execution group)
-l specifies the level of trace (in this case, none)

On z/OS, enter the command `F MQP1BRK,ct u=yes, e='default', l=none`

Example: stopping user trace on a message flow from the workbench

To stop trace for one of your message flows from the workbench:

1. Switch to the Broker Administration perspective.
2. In the Domains view, right-click the message flow for which you want to stop trace.
3. Click **User trace** → **None**.

Retrieving user trace

Use the **mqsireadlog** command to access the trace information recorded by the user trace facilities. This command retrieves the trace details according to parameters that you specify on the command, and writes the requested records to a file, or to the command line window, in XML format.

Example: retrieving user trace information in XML format

To retrieve information for the user trace activated with the **mqsichangetrace** command and write it to an output file, on distributed platforms, enter the command `mqsireadlog WBRK_BROKER -u -e default -o trace.xml` where:

-u specifies user trace
-e specifies the execution group (in this case, the default execution group)
-o specifies the output file (in this case, trace.xml)

This sends a log request to the broker to retrieve the user trace log, and stores the responses in the trace.xml file. You can view this file using a plain text editor.

Starting service trace

Activate service traces only when you receive an error message that instructs you to start service trace, or when directed to do so by your IBM Support Center.

Use the **mqsichangetrace** command to start WebSphere Message Broker service trace facilities.

You can select only one broker on each invocation of the command, but you can activate concurrent traces for more than one broker by invoking the command more than once.

You must specify the individual component that you want to trace; for example the broker, Configuration Manager, or User Name Server, to limit the scope of a trace.

If you want to trace the command executable files themselves, set the environment variables MQSI_UTILITY_TRACE and MQSI_UTILITY_TRACESIZE before you initiate trace. Ensure that you reset these variables when the command that you are tracing has completed. If you do not do so, all subsequent commands are also traced, and their performance is therefore degraded.

If the trace cannot be associated with a specific component, the component name part of the file name is set to `utility`, for example, when tracing a command such as **mqsilist**, when no arguments are used.

In order to trace the **mqsideploy**, **mqsistartmsgflow**, **mqsistopmsgflow**, **mqsicreateexecutiongroup**, and **mqsideleteexecutiongroup** commands, use the **-v** flag. This takes an argument that is the name of the file to which trace will be written.

Example: starting service trace for the User Name Server

To start debug level service tracing for the User Name Server, on distributed platforms, enter the command `mqsichangetrace UserNameServer -t -b -l debug` where:

-t specifies service trace
-b specifies that trace for the `agent` subcomponent of the specified component is to be started
-l specifies the level of trace (in this case, debug)

On z/OS, enter the command `F MQPIUNS,ct t=yes, b=yes, l=debug`

Checking service trace options

Use the **mqsireporttrace** command to check what tracing options are currently active for your brokers.

Specify the component for which the check is required. The command responds with the current trace status for the component that you have specified.

Example: checking service trace options for a broker

To check what options are currently set for the User Name Server, on distributed platforms, enter the command `mqsireporttrace UserNameServer -t` where **-t** specifies service trace.

On z/OS, enter the command `F MQP1UNS,reporttrace t=yes`

If you have started tracing by following the example in "Starting service trace" on page 112, the response to the **mqsireporttrace** command is:

```
BIP8098I: Trace level: debug, mode: safe, size: 1024 KB
BIP8071I: Successful command completion
```

Changing service trace options

Use the **mqsichangetrace** command to change the service trace options that you have set.

Example: changing service trace from debug to normal

To change from a debug level of trace to a normal level on the User Name Server, on distributed platforms, enter the command `mqsichangetrace UserNameServer -t -b -l normal` where:

-t specifies service trace
-b specifies that tracing for the `agent` subcomponent of the specified component is to be changed
-l specifies the level of trace (in this case, changing it to normal)

On z/OS, enter the command `F MQP1UNS,ct t=yes, b=yes, l=normal`

Stopping service trace

Use the **mqsichangetrace** command with a trace level of `none` to stop an active trace. This stops the trace activity for the component that you specify on the command. It does not affect active traces on other components. For example, if you stop tracing on the execution group `test`, an active trace on another execution group continues.

If you redeploy a component from the workbench, trace for that component is returned to its default setting of `none`.

Example: stopping service trace on the User Name Server

To stop the trace started by the command shown in “Starting service trace” on page 112, on distributed platforms, enter the command `mqsichangetrace UserNameServer -t -b -l none` where:

-t specifies service trace
-b specifies that trace for the `agent` subcomponent of the specified component is to be stopped
-l specifies the level of trace (in this case, none)

On z/OS, enter the command `F MQP1UNS,ct t=yes, b=yes, l=none`

Retrieving service trace

Use the **mqsireadlog** command to access the trace information recorded by the service trace facilities.

This command retrieves the trace details according to parameters that you specify on the command, and writes the requested records to a file, or to the command line window, in XML format.

Example: retrieving service trace information in XML format

To retrieve information for the service trace activated with the **mqsichangetrace** command, and write it to an output file, on distributed platforms, enter the command `mqsireadlog UserNameServer -t -b agent -f -o strace.xml` where:

-t specifies service trace
-b *agent* specifies that trace for the `agent` subcomponent of the specified component

is to be retrieved
-f specifies that the log file is to be read directly from the file system (this flag is mandatory for service trace)
-o specifies the output file (in this case, strace.xml)

This sends a log request to the broker to retrieve the service trace log, and stores the responses in the file strace.xml. You can view this file using a plain text editor.

Formatting trace

The trace information generated by the **mqsireadlog** command is not easy to read unless you use an XML viewer (such as an Internet browser) or an XML editor that understands the document type descriptor (DTD) in the file.

WebSphere Message Broker provides the command **mqsiformatlog** to format the trace information to a flat file, so that you can view it using a text editor.

The **mqsiformatlog** command takes a file generated by the **mqsireadlog** command as input, and flattens the XML log into structured records. It also retrieves the inserts for the XML message in your current locale. You can direct the formatted output to a file, or view it in the command line window.

Each trace entry contains a time stamp and a WebSphere Message Broker message that contains a number (for example, BIP2622) and a text string containing variable inserts.

Example: formatting user trace information on Windows

On Windows, to format the trace file created in “Starting user trace” on page 109, enter the command `mqsiformatlog -i trace.xml -o formattrace.log` where:

-i specifies the input file (in this case, trace.xml)
-o specifies the output file (in this case, formattrace.log)

This command reads the trace information in the file trace.xml, formats it, and writes it to the file formattrace.log. The following example shows a portion of the output of the **mqsiformatlog** command for a normal level trace file.

```
Timestamps are formatted in local time, 330 minutes past GMT.

2003-06-19 11:30:29.795999     2852   UserTrace   BIP2632I: Message received and propagated
 to 'out' terminal of MQ Input node 'Video_Test.VIDEO_XML_IN'.
2003-06-19 11:30:29.795999     2852   UserTrace   BIP6060I: Parser type 'Properties' created
 on behalf of node 'Video_Test.VIDEO_XML_IN' to handle portion of incoming message of length
 0 bytes beginning at offset '0'.
2003-06-19 11:30:29.795999     2852   UserTrace   BIP6061I: Parser type 'MQMD' created on
 behalf of node 'Video_Test.VIDEO_XML_IN' to handle portion of incoming message of length
 '364' bytes beginning at offset '0'. Parser type selected based on value 'MQHMD' from
 previous parser.
2003-06-19 11:30:29.795999     2852   UserTrace   BIP6061I: Parser type 'MRM' created on
 behalf of node 'Video_Test.VIDEO_XML_IN' to handle portion of incoming message of length
 '650' bytes beginning at offset '364'. Parser type selected based on value 'MRM' from
 previous parser.
2003-06-19 11:30:29.795999     2852   UserTrace   BIP2537I: Node 'Video_Test.Extract Fields':
 Executing statement 'BEGIN ... END;' at (.Video_Test_Compute.Main, 2.2).
2003-06-19 11:30:29.795999     2852   UserTrace   BIP2537I: Node 'Video_Test.Extract Fields':
 Executing statement 'SET OutputRoot = InputRoot;' at (.Video_Test_Compute.Main, 7.3).
2003-06-19 11:30:29.795999     2852   UserTrace   BIP2538I: Node 'Video_Test.Extract Fields':
 Evaluating expression 'InputRoot' at (.Video_Test_Compute.Main, 7.20).
2003-06-19 11:30:29.795999     2852   UserTrace   BIP2568I: Node 'Video_Test.Extract Fields':
 Performing tree copy of 'InputRoot' to 'OutputRoot'.

...

2003-06-19 11:30:29.827999     2852   UserTrace   BIP4124I: Message propagated to 'out'
 terminal of Compute node 'Video_Test.Extract Fields'.
2003-06-19 11:30:29.827999     2852   UserTrace   BIP2638I: The MQ Output node
 'Video_Test.VIDEO_OUT' attempted to write a message to queue 'VIDEO_OUT' connected
 to queue manager ''. The MQCC was '0' and the MQRC was '0'.
2003-06-19 11:30:29.827999     2852   UserTrace   BIP2622I: Message successfully output
 by output node 'Video_Test.VIDEO_OUT' to queue 'VIDEO_OUT' on queue manager ''.

Threads encountered in this trace:
  2852
```

Interpreting trace

A formatted log file, like the one in “Formatting trace” on page 115, contains a sequence of WebSphere Message Broker messages that records the activity in a specific part of the system (the part that you identify when you start the trace). You can use this sequence to understand what is happening, and to check that the behavior recorded is what you are expecting.

For example, if you have activated message flow trace, you can see entries that record the path that a message takes through the message flow, and why decisions result in this path (where a choice is available).

If you see unexpected behavior in a message flow, or execution group, you can use this trace information to check the actions taken and identify the source of an error or other discrepancy.

The messages contain identifiers for the resources that are being traced, for example the execution groups and message flows. The identifier given is usually the label (the name) that you gave the resource when you defined it.

Here is an extract from a user trace file. In the example, each column has been labelled:

```
Timestamp                  Thread ID Trace type  Message
2005-07-12 16:17:18.242605   5344   UserTrace   BIP2537I: Node 'Reply.MapToRequestor':
                                                Executing statement   ''SET I = I + 1;''
                                                at ('.MapToRequestor.CopyMessageHeaders',
                                                '6.4').
2005-07-12 16:17:18.242605   5344   UserTrace   BIP2539I: Node 'Reply.MapToRequestor':
                                                Evaluating expression ''I'' at
                                                ('.MapToRequestor.CopyMessageHeaders',
                                                '6.12'). This resolved to ''I''. The
                                                result was ''1''.
2005-07-12 16:17:18.242605   5344   UserTrace   BIP2539I: Node 'Reply.MapToRequestor':
                                                Evaluating expression ''I + 1'' at
                                                ('.MapToRequestor.CopyMessageHeaders',
                                                '6.14'). This resolved to ''1 + 1''.
                                                The result was ''2''.
2005-07-12 16:17:18.242605   5344   UserTrace   BIP2566I: Node 'Reply.MapToRequestor':
                                                Assigning value ''2'' to field / variable
                                                ''I''.
```

References such as `'6.12'` apply to the row and column number within a function that specify the location of the command that is being executed; in this case, row 6, column 12.

Clearing old information from trace files

If the component that you are tracing has stopped, you can delete its trace files from the log subdirectory of the WebSphere Message Broker home directory.

If you are tracing an execution group, you can use the **-r** parameter of the **mqsichangetrace** command to reset (clear) the trace log (the **-r** parameter can be specified only if you specify the **-e** parameter). You might do this, for example, when you start a new trace to ensure that all the records on the log are unique to the new trace.

Example: clearing the user trace log for the default execution group

To clear the user trace log for the default execution group, on distributed systems, enter the command: **mqsichangetrace WBRK_BROKER -u -e default -r** where:

WBRK_BROKER specifies the name of the broker
-u specifies user trace
-e specifies the execution group (in this case the default execution group)
-r clears the trace log

On z/OS, enter the command `F MQP1BRK,ct u=yes, e='default', r=yes`

Changing trace settings from the workbench

In addition to collecting user trace and service trace for various components, you can also collect the following additional types of trace from the workbench:

- Trace for all WebSphere Message Broker components, including the workbench itself. This uses the logger.
- Configuration Manager proxy trace
- Broker Administration trace

The following sections tell you how to change the settings for these types of trace.

Changing the logger settings

WebSphere Message Broker provides a logger that controls tracing for all the components of WebSphere Message Broker. This is the com.ibm.etools.mft.util plug-in. By default, the logger is set to produce trace for severe errors, but you can change this level:

1. Click **Windows** → **Preferences**.
2. Click **Logging** from the left menu.
3. Click **com.ibm.etools.mft.util**.
4. Click the **General** tab.
5. Open the drop-down list for the Default logging level, and select the logging level that you want. For the most detailed report, select **FINEST**.
6. Click **OK**.

The files from this trace are written to the directory *install_dir*\eclipse\workbench\.metadata. Logging utility output is written to the LoggingUtil.log file. Any workbench errors are written to the .log file.

Changing the Configuration Manager proxy trace settings

You control tracing of the Configuration Manager proxy from the workbench:

1. Click **Windows > Preferences**.
2. Expand **Broker Administration** in the left menu.
3. Click **Configuration Manager proxy**.
4. In the Configuration Manager proxy tracing section of the menu, change the level of tracing that is performed. The default is None. If you select **Errors to file**, the default file that trace is written to is *install_dir*\eclipse\workspace\cmptraceerror.log.
5. Click **OK**.

Changing the Broker Administration trace settings

You control tracing of the Broker Administration from the workbench:

1. Click **Windows** → **Preferences**.
2. Expand **Broker Administration** in the left menu.
3. Click **Trace**.
4. Change the type of tracing. The default is None. If you select **Stored in a file**, the default file to which trace is written is *install_dir*/eclipse/workspace/.metadata/.plugins/com.ibm.etools.mft.admin/mbda.log.
5. Click **OK**.

ODBC trace

There are various ways that you can initiate trace for ODBC activity, depending on the operating system that you are using:

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, go to the same panel and click **Stop Tracing Now**, then click **OK**.

For Linux and UNIX operating systems, edit the ODBC.ini file to initiate trace for ODBC activity (if you are using the default ODBC.ini file, located at /var/wmqi/odbc/.odbc.ini):

1. Under the stanza entry `[ODBC]` change `Trace=0` to `Trace=1`
2. Optionally, modify the TraceFile file path value to a preferred value. All trace records go to one file, which can be confusing if you have multiple flows or execution groups.

For z/OS, edit /component_HFS/dsnaoini to initiate application trace for ODBC activity:

1. Under the stanza `[COMMON]` change `APPLTRACE=0` to `APPLTRACE=1`

2. Optionally, modify the APPLTRACEFILENAME file path to a preferred value. All trace records go to one file, which can be confusing if you have multiple flows or execution groups.

Trace information is output to the location specified in the .odbc.ini file, or in the ODBC configuration on Windows. By default, this is /var/wmqi/odbc/odbctrace.out on non-Windows.

WebSphere MQ Java Client trace

You can activate WebSphere MQ Java Client trace while putting a message on a WebSphere MQ queue using the Enqueue editor.

1. Click **Window** → **Preferences**.
2. Click **Enqueue** on the left. The Enqueue preferences screen opens.
3. To activate WebSphere MQ Java Client tracing, click the **To file** radio button. The default output file is specified. You can specify a different output file by typing in the filepath or using the **Browse** button.

   If the output file that you specify is invalid (if it is read-only or already in use), you will get error message BIP832:

   ```
   A class java.io.FileNotFoundException exception occurred which reported the following
   message: [filepath] (The process cannot access the file because it is being used by another
   process). Resolve the reason of error and try again.
   ```

   If you do not resolve this error, you will get this BIP832 error message the next time you start up if the output file is still invalid.

You can also enable and disable WebSphere MQ Java Client service trace from the **File** menu of the Configuration Manager Proxy API Exerciser.

If trace is activated, any trace messages written using the Enqueue editor are written to the specified output file.

Configuration Manager Proxy trace

You can enable and disable service trace for the Configuration Manager Proxy (CMP).

Enabling Configuration Manager Proxy trace

To enable tracing for the Configuration Manager Proxy for your application, use the following command in your code:

```
// Enable Config Manager Proxy service trace
   ConfigManagerProxy.enableTracing("outputfile.txt");
```

This logs all calls to the Configuration Manager Proxy to the outputfile.txt file in the current directory. All Configuration Manager Proxy activity in the entire Java Virtual Machine is logged.

You can also enable Configuration Manager Proxy service trace from the **File** menu of the Configuration Manager Proxy API Exerciser.

In addition, because the Configuration Manager Proxy uses the MQ Java client, you can enable MQ Java client tracing.

Disabling Configuration Manager Proxy trace

To disable tracing for the Configuration Manager Proxy for your application, use the following command in your code:

```
// Disable Config Manager Proxy service trace
   ConfigManagerProxy.disableTracing();
```

You can also disable Configuration Manager Proxy service trace from the **File** menu of the Configuration Manager Proxy API Exerciser.

Using dumps

This section contains the following topics:

- "Checking for dumps"
- "Using the DUMP command on z/OS"

Checking for dumps

If a dump or abend occurs on your system, an error message is produced.

- **On Windows**

  BIP2111 error message (message broker internal error). The error message contains the path to the MiniDump file in your errors directory.
- **On UNIX**

  BIP2060 error message (execution group terminated unexpectedly). Look in the directory where the broker was started, or in the service user ID's home directory, to find the core dump file.
- **On z/OS**
  - Abend X'2C3' followed by a reason code, for example:

    ```
    IEF450I STU1051 - ABEND=S2C3 U0000 REASON=01050001
    ```

    Look in the system's dump directory for the dump file.
  - An entry in the z/OS syslog for a SYSMDUMP. Look in the started task's user directory for the coredump.pid file, as specified in the z/OS syslog.

Using the DUMP command on z/OS

You might be asked to dump any or several of the following address spaces for IBM to resolve the problem:

- Control address space
- DataFlowEngine address space
- UserNameServer address space
- OMVS kernel address space

The following procedure demonstrates how to dump the DataFlowEngine address space. This procedure is the same for any of the address spaces.

1. Find the address space ID of the address space that you want to dump using the display command on the z/OS syslog:

   ```
   D OMVS,U=your started task user ID
   ```

   This command displays the address spaces of all the processes running from your started task user ID, for example:

```
D OMVS,U=MQ01BRK
BPXO040I 16.14.30 DISPLAY OMVS 237
OMVS     000D ACTIVE          OMVS=(14)
USER     JOBNAME  ASID        PID       PPID STATE   START     CT_SECS

MQ01BRK  MQ01BRK  009D   67306064   84083282 HRI--- 15.41.55     48.37
  LATCHWAITPID=         0 CMD=bipservice MQ01BRK AUTO
MQ01BRK  MQ01BRK  009D   84083282          1 1WI--- 15.41.55     48.37
  LATCHWAITPID=         0 CMD=/argoinst/S000_L30307_P/usr/lpp/mqsi/bin
MQ01BRK  MQ01BRK  009D   16974444   67306064 HRI--- 15.42.01     48.37
  LATCHWAITPID=         0 CMD=bipbroker MQ01BRK
MQ01BRK  MQ01BRK  009F   16974445          1 1W---- 15.42.05   2914.22
  LATCHWAITPID=         0 CMD=/argoinst/S000_L30307_P/usr/lpp/mqsi/bin
MQ01BRK  MQ01BRK  009F   33751662   16974445 HR---- 15.42.05   2914.22
  LATCHWAITPID=         0 CMD=DataFlowEngine MQ01BRK ca614eec-f300-000
```

The infrastructure main program bipimain is the first process in every address space. For a control address space, bipimain starts the bipservice process, which starts the bipbroker process, which might also start the biphttplistener process, depending on the configuration. For a DataFlowEngine address space, bipimain starts the DataFlowEngine process. For a UserNameServer address space, bipimain starts the bipuns process. For each execution group, an additional DataFlowEngine address space is started. In this example, only one execution group is deployed.

2. Use the z/OS DUMP command to dump the DataFlowEngine address space, which is shown in the above example as 9F.
   a. Enter the following command:

      ```
      DUMP TITLE=(DFE)
      ```

      The console returns:

      ```
      *`15 IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
      ```

   b. Enter:

      ```
      R 15,ASID=9F,CONT
      ```

      The console returns:

      ```
      *16 IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
      ```

   c. Enter:

      ```
      R 16,SDATA=(CSA,RGN,PSA,SQA,LSQA,LPA,TRT,GRSQ,SUM),END
      ```

      The console returns:

      ```
      IEE600I REPLY TO 16 IS;SDATA=(CSA,RGN,PSA,SQA,LSQA,LPA,TRT,GRSQ,SUM),END
      IEA794I SVC DUMP HAS CAPTURED: 356
      DUMPID=014 REQUESTED BY JOB (*MASTER*)
      DUMP TITLE=DFE
      IEF196I IGD101I SMS ALLOCATED TO DDNAME (SYS00018)
      IEF196I         DSN (SYS3.DUMP.ARG5.#MASTER#.T142958.S00014      )
      IEF196I         STORCLAS (SMS) MGMTCLAS (DUMP) DATACLAS (        )
      IEF196I         VOL SER NOS= ARGSMR
      ```

      The dump is stored in either a pre-allocated dump data set called SYS1.DUMPxx, or an automatically allocated dump data set named according to an installation-specified pattern.

3. In some scenarios, all address spaces for a given broker, that is, all those listed in the example above, plus the OMVS address space and dataspaces, are needed by IBM to resolve the problem. Use the z/OS DUMP command to dump all MQ01BRK address spaces.
   a. Enter the command:

```
DUMP TITLE=(ALL)
```

The console returns:

```
`15 IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
```

b. Enter:

```
R 15,JOBNAME=(OMVS,MQ01BRK),DSPNAME=('OMVS'.*),SDATA=(PSA,SQA,LSQA,RGN,TRT,
  LPA,CSA,GRSQ,SUM,NUC)
```

The console returns:

```
IEE600I REPLY TO 15 IS;JOBNAME=(OMVS,MQ01BRK),DSPNAME=('OMVS'.*),S
IEA794I SVC DUMP HAS CAPTURED: 303
DUMPID=040 REQUESTED BY JOB (*MASTER*)
DUMP TITLE=ALL
IEE853I 13.40.40 SYS1.DUMP TITLES 306
SYS1.DUMP DATA SETS AVAILABLE=000 AND FULL=000
CAPTURED DUMPS=0001, SPACE USED=00000447M, SPACE FREE=00001053M
    DUMP.MVK4.#MASTER#.D030415.T134007.S00039 DATA UNAVAILABLE WHILE
        BEING DUMPED TO
IEA611I COMPLETE DUMP ON DUMP.MVK4.#MASTER#.D030415.T134007.S00039 309
DUMPID=040 REQUESTED BY JOB (*MASTER*)
FOR ASIDS(000D,009D,009F)
```

You can also find information on the individual thread by issuing the DISPLAY z/OS console command, as in the example:

```
D OMVS,PID=83886535
```

Recovering after failure

The types of recovery described in this section are really a last resort. You must try to get to the root of the problem first, using the diagnosis techniques described throughout the "Troubleshooting and support" on page 3 section of the information center. If your problem cannot be resolved using these techniques, contact your IBM Support Center.

This section contains the following topics:

- "Recovering after the Configuration Manager fails"
- "Recovering after the broker fails" on page 123
- "Recovering after the User Name Server fails" on page 124
- "Recovering after an execution group fails" on page 124
- "Recovering after the broker's queue manager fails" on page 125
- "Recovering after the Configuration Manager's queue manager fails" on page 125
- "Recovering after the User Name Server's queue manager fails" on page 127
- "Recovering after a shared queue manager fails" on page 127
- "Recovering after the configuration repository fails" on page 128
- "Recovering after the broker database fails" on page 129

Recovering after the Configuration Manager fails

Before you start:

Try to get to the root of the problem first, using the diagnosis techniques described throughout the "Troubleshooting and support" on page 3 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 the Configuration Manager environment is damaged, or the configuration repository is corrupted, and the problem cannot be corrected using problem determination, or by the IBM Support Center, perform the following sequence of operations to recreate the Configuration Manager.

The following sequence of operations applies to Windows, Linux, and UNIX platforms. For z/OS you must use the JCL equivalents of the commands.

1. Ensure that all workbench sessions are stopped.
2. Stop the Configuration Manager using the **mqsistop** command.
3. Stop the Configuration Manager's queue manager using the **endmqm** command.
4. Delete the Configuration Manager using the **mqsideleteconfigmgr** command:
   a. If you are recovering the Configuration Manager because the configuration repository is damaged, delete it when you delete the Configuration Manager. This destroys all information pertinent to the broker domain (not just data internal to the Configuration Manager). You must export everything you can from the domain before you issue the **mqsideleteconfigmgr** command specifying the **-n** flag.
   b. If the configuration repository is undamaged, omit the **-n** flag. This preserves your configuration data in the repository.

   The **mqsideleteconfigmgr** command:
   - Stops the service that runs the Configuration Manager.
   - Stops and deletes the WebSphere MQ queue manager for the Configuration Manager, if requested.
   - Removes the tables from the configuration repository, if requested.
   - Removes the record for the component in the broker registry.
5. If you are recovering the Configuration Manager because the configuration repository is damaged, restore the repository from a previously successful backup version using the **mqsirestoreconfigmgr** command.
6. Recreate the Configuration Manager using the **mqsicreateconfigmgr** command.
7. Start the Configuration Manager using the **mqsistart** command.
8. Restart the workbench if it is not currently running, and switch to the Broker Administration perspective.
9. If you have completed step 4a and step 5, you must also redeploy the domain configuration to ensure that the configuration across the broker domain is consistent.

Recovering after the broker fails

Before you start

Try to get to the root of the problem first, using the diagnosis techniques described throughout the "Troubleshooting and support" on page 3 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 a broker's database becomes corrupted, follow the instructions described in "Recovering after the broker database fails" on page 129. If the database is not corrupted but the problem cannot be corrected using problem determination, perform the following sequence of operations to re-create the broker:

1. Ensure that no workbench users are deploying to brokers. You must wait until these actions have completed.
2. Stop the broker using the mqsistop command.
3. Stop the broker's queue manager using the **endmqm** command.
4. Take a backup of the broker database tables. These tables are interdependent and must all be in a consistent state when restored. You cannot backup or restore individual tables.
5. Delete the broker using the mqsideletebroker command.
6. Re-create the broker using the mqsicreatebroker command.
7. Restore the broker database tables, either from the backup you have just taken, or from a previously successful backup version.
8. Start the broker using the mqsistart command.
9. Redeploy the domain configuration to ensure that the configuration across the broker domain is consistent.

Recovering after the User Name Server fails

Before you start

Try to get to the root of the problem first, using the diagnosis techniques described throughout the "Troubleshooting and support" on page 3 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 the User Name Server environment becomes damaged, and the problem cannot be corrected using problem determination, or by the IBM Support Center, perform the following sequence of operations to recreate the User Name Server:

1. Stop the User Name Server using the **mqsistop** command.
2. Stop the User Name Server's queue manager using the **endmqm** command.
3. Delete the User Name Server using the **mqsideleteusernameserver** command.
4. Re-create the User Name Server using the **mqsicreateusernameserver** command.
5. Start the User Name Server using the **mqsistart** command.

Recovering after an execution group fails

Before you start

Try to get to the root of the problem first, using the diagnosis techniques described throughout the "Troubleshooting and support" on page 3 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 a single execution group fails, and the problem cannot be corrected using problem determination, or by the IBM Support Center, perform the following sequence of operations to recreate the execution group:

1. Ensure that no workbench users are deploying to the broker to which the execution group belongs. You must wait until these actions have completed.
2. Stop the broker using the mqsistop command.
3. Delete the execution group.
4. Start the broker using the mqsistart command.
5. Clean the execution group.
6. Redeploy the configuration.

If more than one execution group fails, you might have to recreate the broker. See "Recovering after the broker fails" on page 123 for information on how to do this.

Recovering after the broker's queue manager fails

Before you start

Try to get to the root of the problem first, using the diagnosis techniques described throughout the "Troubleshooting and support" on page 3 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 the failed queue manager is shared between a broker, User Name Server and Configuration Manager, re-create all three components, because each component creates different queues. For information on how to recover a shared queue manager, see "Recovering after a shared queue manager fails" on page 127.

If the broker's queue manager fails and cannot be corrected 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 workbench users are deploying to the broker. You must wait until any such actions have completed.
2. Stop the broker using the **mqsistop** command.
3. If there is no damage to any of the broker database tables, take a backup of these tables. These tables are interdependent and must all be in a consistent state when restored. You cannot back up or restore individual tables.
4. Delete the broker using the **mqsideletebroker** command, with the **-q** flag to remove the queue manager.
5. Re-create the broker using the **mqsicreatebroker** command. The **mqsicreatebroker** command creates the queue manager and default queues automatically.
6. Re-create any specific queues that are needed for your message flows.
7. Restore the broker database tables, either from the backup you have just taken, or from a previously-successful backup version.
8. Start your brokers using the **mqsistart** command.
9. Redeploy the domain configuration.

Recovering after the Configuration Manager's queue manager fails

Before you start

Try to get to the root of the problem first, using the diagnosis techniques described throughout the "Troubleshooting and support" on page 3 section of the information center. If your problem cannot be resolved using these techniques, contact your IBM Support Center.

Follow the link below for your operating system:
- "Windows, Linux, and UNIX systems"
- "z/OS"

If you need to transfer the Configuration Manager onto another queue manager, follow the steps described in Moving the Configuration Manager to a new queue manager.

Windows, Linux, and UNIX systems

Use the procedure in this section only as a last resort.

If the failed queue manager was shared between a broker, User Name Server and Configuration Manager, re-create all three components, because each component creates different queues. For information on how to recover a shared queue manager, see "Recovering after a shared queue manager fails" on page 127.

If the Configuration Manager's queue manager fails and cannot be corrected using problem determination, or by the IBM Support Center, perform the following sequence of operations to recreate the queue manager:
1. Ensure that no workbench users are deploying to brokers. You must wait until any such actions have completed.
2. Stop all brokers that are using the Configuration Manager using the **mqsistop** command.
3. Stop the Configuration Manager using the **mqsistop** command.
4. Delete the Configuration Manager using the **mqsideleteconfigmgr** command, with the **-q** flag to remove the queue manager.
5. Re-create the Configuration Manager using the **mqsicreateconfigmgr** command. The **mqsicreateconfigmgr** command creates the queue manager and default queues automatically.
6. Start the Configuration Manager using the -q command.
7. Start your brokers using the **mqsistart** command.
8. Redeploy the domain configuration.

z/OS

Use the procedure in this section only as a last resort.

If the failed queue manager was shared between a broker, User Name Server and Configuration Manager, re-create all three components, because each component creates different queues. For information on how to recover a shared queue manager, see "Recovering after a shared queue manager fails" on page 127.

If the Configuration Manager's queue manager fails and cannot be corrected using problem determination, or by the IBM Support Center, perform the following sequence of operations to recreate the queue manager:
1. Ensure that no workbench users are deploying to brokers. You must wait until any such actions have completed.

2. Stop all brokers that are using the Configuration Manager.
3. Stop the Configuration Manager.
4. Correct any queue manager problems.
5. Re-create the Configuration Manager queues using the BIPCRCM JCL; see Sample BIPCRCM file. Include the **mqsicreateconfigmgr** command **-2** parameter.
6. Start the Configuration Manager.
7. Start your brokers.
8. Redeploy the domain configuration.

Recovering after the User Name Server's queue manager fails

Before you start

Try to get to the root of the problem first, using the diagnosis techniques described throughout the "Troubleshooting and support" on page 3 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 the failed queue manager was shared between a broker, User Name Server, and Configuration Manager, re-create all three components, because each component creates different queues. For information on how to recover a shared queue manager, see "Recovering after a shared queue manager fails."

If the User Name Server's queue manager fails and cannot be corrected using problem determination, or by the IBM Support Center, perform the following sequence of operations to recreate the queue manager:

1. Ensure that no workbench users are deploying to brokers. You must wait until any such actions have completed.
2. Stop the User Name Server using the **mqsistop** command.
3. Delete the User Name Server using the **mqsideleteusernameserver** command, with the **-q** flag to remove the queue manager.
4. Re-create the User Name Server using the **mqsicreateusernameserver** command. The **mqsicreateusernameserver** command creates the queue manager and default queues automatically.
5. Start the User Name Server using the **mqsistart** command.
6. Redeploy the domain configuration.

Recovering after a shared queue manager fails

Before you start

Try to get to the root of the problem first, using the diagnosis techniques described throughout the "Troubleshooting and support" on page 3 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 a queue manager that is shared between a broker, User Name Server, and Configuration Manager fails and cannot be corrected using problem determination, or by the IBM Support Center, re-create all three components, because each component creates different queues. Perform the following sequence of operations:

1. Ensure that no workbench users are deploying to brokers. You must wait until any such actions have completed.
2. Stop each component using the **mqsistop** command.
3. Delete the User Name Server using the **mqsideleteusernameserver**command.
4. Delete the broker using the **mqsideletebroker** command.
5. Delete the Configuration Manager using the **mqsideleteconfigmgr** command, with the **-q** flag to remove the queue manager.

   You must delete the queue manager last. It cannot be deleted while other components are still sharing it.
6. Re-create each component using the relevant mqsi command.
7. Start each component using the **mqsistart** command.
8. Redeploy the domain configuration.

If you need to transfer the Configuration Manager onto another queue manager, follow the steps described in Moving the Configuration Manager to a new queue manager.

Recovering after the configuration repository fails

Before you start

Try to get to the root of the problem first, using the diagnosis techniques described throughout the "Troubleshooting and support" on page 3 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 the configuration repository fails and cannot be corrected using problem determination, or by the IBM Support Center, perform the following sequence of operations to re-create the configuration repository:

1. Ensure that no workbench users are deploying to brokers. You must wait until any such actions have completed.
2. Stop all brokers that are using the Configuration Manager using the **mqsistop** command.
3. Stop the Configuration Manager using the **mqsistop** command.
4. Re-create the configuration repository from a backup version using the **mqsirestoreconfigmgr** command on the same machine as the Configuration Manager.

   If you do not have a backup of your configuration repository, re-create the Configuration Manager at this stage. This re-creates the configuration repository. However, any of the data that you previously had in the repository will be lost. To re-create the Configuration Manager:
   a. Delete the Configuration Manager using the **mqsideleteconfigmgr** command(**specifying the -n parameter)**.
   b. Re-create the Configuration Manager using the **mqsicreateconfigmgr** command. The **mqsicreateconfigmgr** command creates the configuration repository automatically.
5. Start the Configuration Manager using the **mqsistart** command.
6. Start your brokers using the **mqsistart** command.

Recovering after the broker database fails

Before you start

Try to get to the root of the problem first, using the diagnosis techniques described throughout the “Troubleshooting and support” on page 3 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 the broker database fails and cannot be corrected using problem determination, or by the IBM Support Center, perform the following sequence of operations to re-create the broker:

1. Ensure that no workbench users are deploying to the broker. You must wait until any such actions have completed.
2. Stop the broker using the **mqsistop** command.
3. If possible, re-create the broker database from a backup version. If this is not possible, perform the following actions in the order shown:
   a. Delete the broker using the **mqsideletebroker** command.
   b. Remove completely all references to the broker from the Configuration Manager repository. You can do this by removing the broker from the topology using the Message Brokers Toolkit. Ensure that the broker’s SYSTEM.BROKER.ADMIN.QUEUE queue has been deleted and initiate a topology deploy.

      Alternatively, you can use the Configuration Manager Proxy API ″TopologyProxy.removeDeletedBroker(String)″ to programmatically purge all references to the broker:
      1) Start the Configuration Manager Proxy API Exerciser sample (which is available on the **Start** menu on Windows).
      2) Connect to the Configuration Manager.
      3) Right-click the topology object, then click **Remove references to a previously deleted broker**.
   c. Re-create the broker using the **mqsicreatebroker** command.
   d. Add the broker to the Configuration Manager’s domain topology.
4. Start the broker using the **mqsistart** command.
5. If you were unable to restore the broker’s database from a backup, redeploy the broker’s configuration.

Contacting your IBM Support Center

If you cannot resolve problems that you find when you use WebSphere Message Broker, or if you are directed to do so by an error message generated by WebSphere Message Broker, you can request assistance from your IBM Support Center.

Before you contact your Support Center, use the checklist below to gather important information. Some items might not necessarily be relevant in every situation, but you should provide as much information as possible to enable the IBM Support Center to re-create your problem.

For WebSphere Message Broker:

- The product version.

- Any fix packs applied.
- Any interim fixes applied.
- All current trace and error logs, including relevant Windows Event log or Linux and UNIX operating system syslog entries, and any abend or dump files from the *install_dir*\errors directory on Windows, or the /var/mqsi/errors directory on Linux and UNIX. Obtain user trace log files at debug level for all relevant message flows and preferably format them. Also include any requested service trace files.

  To send files from distributed operating systems, create a zipped file using any zip utility.

  To send a file from the HFS to IBM, use **tar** to compress the file. For example `tar -cx -f coredump.0002009E coredump.toibm`. To send MVS data sets to IBM, terse them using `TRSMAIN`, which you can download from z/OS tools download.
- A list of the components installed. Include details of the number of computers and their operating systems, the number of brokers and the computers on which they are running, and the existence and details of any User Name Servers.
- The zipped file obtained by exporting your workspace and appropriate message flows and message sets. This action is performed from the workbench.
- Details of the operation that you were performing, the results that occurred, and the results you were expecting.
- A sample of the messages being used when the problem arose.
- If relevant, the report file from the C or COBOL importer. This is located in the directory from which the file import was attempted.
- If you are using tagged delimited wire format on message sets, the TDS log files.

For WebSphere MQ:

- The product version.
- Any fix packs applied.
- Any interim fixes applied.
- All current trace and error logs, including relevant Windows Event log or Linux and UNIX operating system syslog entries and First Failure Support Technology™ (FFST™) output files. You can find these files, which have the extension .fdc, in the errors subdirectory within the WebSphere MQ home directory.
- Details of WebSphere MQ client software, if appropriate.

For each database that you are using:

- The product and release level (for example, DB2 7.1).
- Any fix packs applied.
- Any interim fixes applied.
- All current trace and error logs, including relevant Windows Event log or Linux and UNIX operating system syslog entries, for example the db2dialog.log file on DB2. Check the database product documentation for details of where to find these files.
- Definitions of any database tables.
- Any ODBC traces.

For Windows:

- The version.
- The Service Pack level.
- The version of the system files msvcrt.dll, msvcp60.dll, msvcirt.dll, and mfc42.dll. You can find these files in the Windows\SYSTEM32 directory. Use the Windows Explorer file properties to display the versions.
- The environment settings.

For Linux and UNIX® operating systems:

- The product version. You can find the version installed by using the **uname -a**command.
- Any service level and patches applied.
- The environment settings.

For z/OS:

- The product version.
- The list of PTFs applied.
- The environment settings.
- The joblogs from all address spaces.

Part 2. Reference

Troubleshooting

The following reference information is provided to help you to diagnose errors in WebSphere Message Broker:

- “Logs”
- “Trace” on page 137
- Trace logging from a user-defined C extension
- “Dumps” on page 139
- “Abend files” on page 142
- “Event reports” on page 144
- “WebSphere MQ facilities” on page 157
- “Database facilities” on page 157
- “Other sources of diagnostic information on z/OS” on page 158
- “Solutions to similar problems” on page 159
- Diagnostic messages

Logs

Logs are one of the first places to look when you are trying to diagnose any type of error. Your system will have a number of different types of logs, and any number of them might be used during operation of WebSphere Message Broker.

The information that is recorded in a log typically consists of a time stamp indicating when the error occurred, and a brief description of the error that can be expanded to provide more details of the error. Sometimes there might be a general error that was caused by a number of other errors, and they are grouped under the general error.

When an error occurs, check the local error log first. The local error log records 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

There are also a number of logs specific to WebSphere Message Broker, which 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 workbench.

Local error logs

WebSphere Message Broker components use the local error log (also known as the system log) to record information about major activities within the system. When an error occurs, check the local error log first.

On Windows, the local error log is the Windows Event log (Application view).

On UNIX and Linux systems, the local error log is the syslog. Where syslog messages are sent depends on how you configure your UNIX or Linux system.

On z/OS, the local error log is the operator console.

Entries in the local error log that are generated by WebSphere Message Broker are identified as follows:

- On Windows, by events from the source `WebSphere Broker v****` and a message number of the form `nnnn`, where `****` is the current four-digit product version number.
- On UNIX, by a message number of the form `BIPnnnn`.
- On z/OS, by a started task ID, and a message number of the form `BIPnnnn`.

When a broker encounters an error, more than one message might be written to the local error log. Typically these start with a general message (for example, `Could not process a message`), and further messages give more details about the cause of the error (for example, `An error was detected while processing the following SQL statement...`).

WebSphere Message Broker logs

This section describes the logs that are specific to WebSphere Message Broker. They are:

- "Eclipse error log"
- "The broker domain Event Log"
- "The Tasks view" on page 137
- "TDS log" on page 137

Eclipse error log

The Eclipse error log captures internal errors from Eclipse or your code. This is useful at run time, because you are likely to cause exceptions during development. As errors are triggered, they are added to the Error Log view of the Plug-in Development perspective, and you can examine details by double-clicking on them. You can also clear the view when you have fixed all the errors, although you cannot clear individual errors. Entries are sorted in reverse chronological order, that is, most recent first. You can bring up the Properties dialog to see entry details.

Some errors might be hierarchical, that is, there might be a general entry for a complex problem and a number of child entries listing the individual problems in detail.

The same information that is shown in the Error Log view is stored in the `.log` file in the *install_dir*/eclipse/workspace/.metadata directory.

The Eclipse error log shows errors generated by the person using that particular workbench.

The broker domain Event Log

Broker domain log information is written to the broker domain Event Log. This log is stored and managed by the Configuration Manager. The Event Log displays messages about events that occur within the broker domain, such as deploying topology or topics hierarchy configuration. The messages can be information, errors, or warnings. You view the messages in the editor area of the Broker Administration perspective.

The broker domain Event Log shows errors generated by all workbench users in that particular domain.

The Tasks view

The Tasks view shows all the outstanding problems in a particular perspective. In the Broker Application Development perspective, the errors in the Tasks view are logic errors in message flows, ESQL, or message sets. In the Broker Administration perspective, the errors in the Tasks view are information about brokers, execution groups, or message flows that are not running. In the Plug-in Development perspective, the errors in the tasks view are code errors.

TDS log

When you use the TDS physical format messages, you must conform to a number of rules that apply to the setting of values of properties. These rules are checked at deploy time, and, if an inconsistency is found, error BIP1836E is displayed in the workbench. Details of the error are written to the TDS.log file in the *install_dir*/log directory.

For more information about the rules for TDS physical format properties, see TDS message model integrity.

Trace

If you cannot get enough information about a particular problem from the entries available in the various logs described in "WebSphere Message Broker logs" on page 136 and "Local error logs" on page 135, the next diagnosis method to consider is using trace. Trace provides more details about what is happening while code executes. 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 workbench.

There are two main types of trace available in WebSphere Message Broker: user trace and service trace. You typically use user trace for debugging your applications; you can trace brokers, execution groups, and deployed message flows. With service trace you can activate more comprehensive broker tracing, and start tracing for the workbench, Configuration Manager, and User Name Server. You can also trace the execution of all the commands described in Commands.

When you set user tracing on, you cause additional processing for every activity in the component that you are tracing. Large quantities of data are generated by the components. Expect to see some impact on performance 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.

User trace

User trace is one of two types of optional trace available in WebSphere Message Broker. User trace is inactive by default, and must be activated explicitly by a command, or by the workbench, when you need more information than that provided by the entries in the Event Log. For more information about logs, see “Logs” on page 135.

Typically you use user trace for debugging your applications, as it can trace brokers, execution groups, and deployed message flows.

You can use the workbench to control most of the trace activity that you need. The workbench also enables you to start and stop tracing on remote systems.

When you activate 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 to see some impact on performance while user 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.

The user trace log files

When trace is active for any component, information is recorded in binary form. The location of the trace logs is as follows:

Windows
If you set the workpath using the **-w** parameter of the mqsicreatebroker command, the location is workpath\log . If you do not specify the broker workpath, the location is C:\Documents and Settings\All Users\Application Data\IBM\MQSI\common\log.

UNIX /var/mqsi/common/log

z/OS /component_HFS/log

The file names reflect the component and subcomponent for which the trace is active. For example, the broker name and unique execution group identifier form part of the file name when you are tracing activity within that execution group.

For example, if you have created a broker called `WBRK_BROKER`, you might see the following files in the log subdirectory:

```
WBRK_BROKER.682ec116-dc00-0000-0080-ce28a236e03d.userTrace.bin.1
WBRK_BROKER.682ec116-dc00-0000-0080-ce28a236e03d.userTrace.bin.2
```

You cannot view these files directly; you must use the commands provided to access the trace information and convert it to a viewable format. Use the **mqsireadlog** command to retrieve the trace log for the specified component. You must issue the command on the computer on which the log was generated. You can specify the output to be directed to a file, which is independent of operating system, and can be transferred to other systems for browsing or formatting using the **mqsiformatlog** command. See mqsireadlog command and mqsiformatlog command for more detailed information.

Using a Trace node

You are recommended to include a Trace node in your message flows when you are developing and testing them. This option not only gives you the ability to trace messages and activity in the flow, but also allows you to specify an alternate target file for the trace contents to isolate the detail in which you are interested. For details of how to use and configure a trace node, see Trace node.

Service trace

Service trace is one of two types of optional trace available in WebSphere Message Broker. Service trace is inactive by default, so you must activate it explicitly using a command when you need more information than that provided by the entries in the Event Log or by user trace.

With service trace you can activate more comprehensive broker tracing, and start tracing for the workbench, Configuration Manager, and User Name Server. You can also trace the execution of all the commands described in Commands, including the trace commands themselves. Use the mqsichangetrace command to work with service trace; you cannot use the workbench.

Activate service trace only when you receive an error message that instructs you to, or when directed to do so by your IBM Support Center.

When you activate service 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 to see some impact in performance while service 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.

The location of the trace logs is as follows:

Windows
If you have set the workpath using the **-w** parameter of the mqsicreatebroker command, the location is workpath\log . If you have not specified the broker workpath, the location is C:\Documents and Settings\All Users\Application Data\IBM\MQSI\common\log.

UNIX /var/mqsi/common/log

z/OS /component_HFS/log

The directory to which the service trace logs are written must be able to hold all the logs for that computer. You might want to place it on a separate file system, if allowed by your system operator.

Dumps

Under exceptional circumstances, for example if errors occur in the broker or User Name Server executable files or in the infrastructure main program, Windows MiniDumps, UNIX core dumps, or z/OS SVC or core dumps might be produced. The amount and complexity of data produced in these cases normally requires the assistance of your IBM Support Center.

Dumps on Windows

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

They are accompanied by a BIP2111 error message (message broker internal error) that contains the path to the MiniDump file in your workpath/errorsdirectory.

MiniDump files have the extension `.dmp`. You can analyze these MiniDumps using a suitable debugger, but we recommend that you get assistance from IBM.

Dumps on UNIX platforms

UNIX core dumps might be produced, in extreme cases, by broker processes. They are never produced during normal operation.

A BIP2060 error message (execution group terminated 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 using a suitable debugger, but we recommend that you get assistance from IBM.

Dumps on z/OS

A WebSphere Message Broker for z/OS broker or User Name Server produces different types of dumps depending on where the original error occurs.

SVC dumps
These are produced for errors in the WebSphere Message Broker for z/OS infrastructure main program (bipimain).

Core dumps
These are produced for errors in the broker or User Name Server executables (bipservice, bipbroker, DataFlowEngine, and bipuns).

SVC dumps

Errors in the infrastructure main program cause system abend X'2C3' followed by a specific reason code. This abend might occur before the WebSphere Message Broker for z/OS recovery routine is active, so might occur without a dump being written. The reason code helps the IBM Support Center to determine the error. In that situation you get only the following information in the z/OS syslog:

```
IEF450I STU1051 - ABEND=S2C3 U0000 REASON=01050001
```

An SVC dump caused by z/OS or WebSphere Message Broker 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.

Under some circumstances, SVC dumps are not produced. Generally, dumps are suppressed because of time or space problems, or security violations. In addition you can suppress SVC dumps that duplicate previous dumps using z/OS DAE, for example, use the command SET DAE=xx. (DAE is Dump Analysis and Elimination: a z/OS service that enables an installation to suppress SVC dumps and ABEND SYSDUMP dumps that are not needed because they duplicate previously written dumps.)

The following example shows a message on the z/OS syslog indicating whether duplicates of SYSMDUMPs 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 SYSMDUMP 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 manual *MVS Diagnosis: Tools and Service Aids* gives details about using z/OS dump analysis and elimination (DAE).

Core dumps

A broker or User Name Server produce USS (UNIX System Services) core dumps. These core dumps are SYSMDUMP dumps. They are written to the started task's user directory. The name is coredump.pid, where `.pid` is the hexadecimal value of the process ID of the process that encountered the error, and is listed on the z/OS syslog. The dump can be found in the started task users' home directory, for example, /u/stu1/coredump.000000ae.

The maximum size of a core dump is defined through MAXCORESIZE in the BPXPRMxx parmlib member. The IBM-supplied default is 4MB. To ensure completeness of a core dump of any WebSphere Message Broker for z/OS address space, change this to 2GB. The started task user's directory must then have at least this size.

When the error recovery routines of WebSphere Message Broker for z/OS provide an SVC dump, a core dump is also written to your started task user ID directory. In this case the SVC dump and the entry on the z/OS syslog are of interest to IBM to resolve the problem. The following is an example of an entry on the z/OS syslog when an SVC dump and core dump have been provided:

```
IEA794I SVC DUMP HAS CAPTURED: 486
 DUMPID=008 REQUESTED BY JOB (STU1051 )
 DUMP TITLE=COMP=WMQI z/OS CID=G9700 ISSUER=BIPIMAIN (BIPIP
            ): WMQI InfraMain           : S0C4-00000004.
 IEF196I IGD101I SMS ALLOCATED TO DDNAME (SYS00010)
 IEF196I         DSN (SYS3.DUMP.ARG5.STU1051.T085940.S00008        )
 IEF196I         STORCLAS (SMS) MGMTCLAS (DUMP) DATACLAS (         )
 IEF196I         VOL SER NOS= ARGSMY
 IEF196I IGD104I SYS3.DUMP.ARG5.STU1051.T085940.S00008       RETAINED,
 IEF196I DDNAME=SYS00010
 IEA611I COMPLETE DUMP ON SYS3.DUMP.ARG5.STU1051.T085940.S00008 493
 DUMPID=008 REQUESTED BY JOB (STU1051 )
 FOR ASID (0060)
 INCIDENT TOKEN: ARGPLEX  ARG5     06/19/2001 08:59:40
  ERROR ID = SEQ02325 CPU00 ASID0060 TIME10.59.40.5
 IEA995I SYMPTOM DUMP OUTPUT 494
 SYSTEM COMPLETION CODE=0C4  REASON CODE=00000004
  TIME=10.59.40  SEQ=02325  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
IEA993I SYSMDUMP TAKEN TO coredump.0300019D
```

To make use of these core dumps, copy them to a partitioned data set 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 as binary. For example:

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

Dump options

To collect all the diagnostic information, 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)

You can check the values on your system by using the D Dump,Options z/OS console command, as follows:

```
D D,O
```

Example output from this command is as follows:

```
SYSABEND- ADD PARMLIB OPTIONS SDATA=(LSQA,TRT,CB,ENQ,DM,IO,ERR,SUM),
                     PDATA=(SA,REGS,LPA,JPA,PSW,SPLS)
SYSUDUMP- ADD PARMLIB OPTIONS SDATA=(LSQA,TRT,CB,ENQ,DM,IO,ERR,SUM),
                     PDATA=(SA,REGS,LPA,JPA,PSW,SPLS)
SYSMDUMP- ADD OPTIONS (NUC,SQA,LSQA,SWA,TRT,RGN,LPA,CSA,SUM,GRSQ)
```

The last entry shows the SYSMDUMP options that are used when taking core dumps.

Abend files

When a process ends abnormally (an abend or dump), an 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 in the broker_work_path\errorsdirectory (if broker_work_path has been set using the -w option of the **mqsicreatebroker** command), or, if the broker work path has not been specified, to the *install_dir*\errors directory, where *install_dir* is the directory where WebSphere Message Broker is installed. If this directory cannot be written to, the file is put in the *install_dir*\log directory, but in either case the message in the Event Log tells you where to find it.

Use the work path option -w of the **mqsicreatebroker** command to create the errors directory in a hard drive partition that does not contain WebSphere Message Broker or Windows itself.

UNIX filename.abend in the /var/wmqi/errors directory.

z/OS filename.abend in the /component_HFS/log directory.

where filename is a unique dynamically-allocated name, as given in the syslog or Event Log message.

Periodically clear any unwanted abend files from the errors or z/OS log directory. In extreme conditions, you might find that your system performance degrades due to significant space being used up with old abend files.

WebSphere Message Broker for z/OS abends

Abend code
2C1

Explanation
The WebSphere Message Broker 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 should refer other codes to the IBM Support Center.

ABN=S2C1-0001xxxx Error opening an HFS file
The HFS component might be full or the broker user ID might not have the correct permissions to access files or directories in the HFS component.

ABN=S2C1-0113xxxx Region size too small

Source
WebSphere Message Broker 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.

Event reports

This topic and the associated topics describe the event messages that are published by a message broker in response to:

- "Configuration changes" on page 145
- "Operational information" on page 150

In addition there are associated topics on:

- "General architecture"
- "Operational warnings" on page 153
- "Notification message schema" on page 154

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 execution group has been created, changed, or deleted
- A message flow has been created, changed, or deleted
- A neighbor has been created, changed, or deleted
- An ACL for a topic has been created, changed, or deleted

Operational information

- A broker has been started or stopped
- A message flow has been started or stopped
- A subscription has been registered or deregistered

Operational warnings

- A retained publication has expired
- A subscription has expired

General architecture

This section describes reserved topics on which brokers publish messages after significant events within the broker. By subscribing to these topics, a client can be informed when 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 should other subsystems (for example, the queue manager or Configuration Manager) that publish system management events through the broker be supported in the future.

Configuration changes

Configuration changes include changes to the operational configuration of a single broker (for example, the addition or removal of a message flow) and changes to the topology for a multibroker network.

This information is described in the following topics:
- "Changes to the local configuration of the broker"
- "Neighbor changes" on page 146
- "ACL updates" on page 148

Changes to the local configuration of the broker

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 execution group for which the configuration has changed.

One such event is published for each configuration request message that is received and processed by an execution group 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 caused the event to be triggered. If an execution group is renamed, subsequent publications that report the state of that execution group 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>
```

Neighbor changes

A change in the set of neighbors (the topology) for a given broker publishes an event using the following system topic:

`$SYS/Broker/`*`broker_name`*`/Neighbor`

where *broker_name* is the name of the broker issuing this message.

The body of each publication is an XML message that describes the change made.

These events are nonpersistent, non-retained publications.

Certain operations delete all neighbor (topology) information at a broker. In this case, the deleted neighbors are not published in the event publication. Instead, the body of the event publication contains a single XML tag that indicates that all neighbors have been deleted.

Examples:

Following are example event messages for a neighbor being created, changed, and deleted, together with an example for all neighbors being deleted.

Neighbor created:

```
Event publication topic = "$SYS/Broker/Broker1/Neighbor"

<Broker uuid="1234" label="Broker1" version="1">
  <ControlGroup>
    <DynamicSubscriptionEngine>
      <Create>
        <Neighbor name="5678" collectiveId="">
          <MQBrokerConnection queueManagerName="nbr_QM_Name"/>
        </Neighbor>
      </Create>    </DynamicSubscriptionEngine>
  </ControlGroup>
</Broker>
```

Neighbor changed:

```
Event publication topic = "$SYS/Broker/Broker1/Neighbor"

<Broker uuid="1234" label="Broker1" version="1">
  <ControlGroup>
    <DynamicSubscriptionEngine>
      <Change>
        <Neighbor name="5678"
          collectiveId="12345678-1234-1234-1234-123456789abc"/>
      </Change>
    </DynamicSubscriptionEngine>
  </ControlGroup>
</Broker>
```

Neighbor deleted:

```
Event publication topic = "$SYS/Broker/Broker1/Neighbor"

<Broker uuid="1234" label="Broker1" version="1">
  <ControlGroup>
    <DynamicSubscriptionEngine>
      <Delete>
        <Neighbor name="5678"/>
      </Delete>
    </DynamicSubscriptionEngine>
  </ControlGroup>
</Broker>
```

All neighbors deleted:

```
Event publication topic = "$SYS/Broker/Broker1/Neighbor"

<Broker uuid="1234" label="Broker1" version="1">
  <ControlGroup>
    <DynamicSubscriptionEngine>
      <Delete>
        <AllNeighbors/>
      </Delete>
    </DynamicSubscriptionEngine>
  </ControlGroup>
</Broker>
```

ACL updates

The creation, deletion, or modification of the ACL associated with a topic causes a publication using the following system topics:

`$SYS/Broker/`*`broker_name`*`/Topic/`*`topic`*

where:

broker_name
is the name of the broker issuing this message.

topic is the topic whose ACL is being modified.

The body of each publication is an XML message that describes the ACL update.

These events are nonpersistent, non-retained publications.

Certain operations delete all ACL entries for a single topic or for all topics. In this case, the individual entries are not published in the event publication. Instead, the body of the event publication contains a single XML tag that indicates that all ACL entries have been deleted.

Examples:

Here are example event messages for an ACL being created, changed, and deleted, together with examples for all ACLs being deleted on a single topic and all topics.

ACL created:

```
Event publication topic = "$SYS/Broker/Broker1/Topic/stock/IBM"

<Broker uuid="1234" label="Broker1" version="1">
  <ControlGroup>
    <DynamicSubscriptionEngine>
      <Topic name="stock/IBM">
        <Create>
          <ACLEntry
            principalName="Fred"
            principalType="user"
            publish="false"
            subscribe="inherit"
            persistent="true"/>
        </Create>
      </Topic>
    </DynamicSubscriptionEngine>
  </ControlGroup>
</Broker>
```

ACL changed:

```
Event publication topic = "$SYS/Broker/Broker1/Topic/stock/IBM"

<Broker uuid="1234" label="Broker1" version="1">
  <ControlGroup>
    <DynamicSubscriptionEngine>
      <Topic name="stock/IBM">
        <Change>
          <ACLEntry
            principalName="Fred"
            principalType="user"
            publish="true"
            subscribe="false"
            persistent="inherit"/>
        </Change>
      </Topic>
    </DynamicSubscriptionEngine>
  </ControlGroup>
</Broker>
```

ACL deleted:

```
Event publication topic = "$SYS/Broker/Broker1/Topic/stock/IBM"

<Broker uuid="1234" label="Broker1" version="1">
  <ControlGroup>
    <DynamicSubscriptionEngine>
      <Topic name="stock/IBM">
        <Delete>
          <ACLEntry principalName="Fred"/>
        </Delete>
      </Topic>
    </DynamicSubscriptionEngine>
  </ControlGroup>
</Broker>
```

All ACLs deleted on a single topic:

```
Event publication topic = "$SYS/Broker/Broker1/Topic/stock/IBM"

<Broker uuid="1234" label="Broker1" version="1">
  <ControlGroup>
    <DynamicSubscriptionEngine>
      <Topic name="stock/IBM">
        <Delete>
          <AllACLEntries/>
        </Delete>
      </Topic>
    </DynamicSubscriptionEngine>
  </ControlGroup>
</Broker>
```

All ACLs deleted on all topics:

```
Event publication topic = "$SYS/Broker/Broker1/Topic"

<Broker uuid="1234" label="Broker1" version="1">
  <ControlGroup>
    <DynamicSubscriptionEngine>
        <Delete>
          <AllACLEntries/>
        </Delete>
    </DynamicSubscriptionEngine>
  </ControlGroup>
</Broker>
```

Operational information

Changes to the processing state of a broker or an individual message flow publish events using the following system topics:

`$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 execution group that contains the message flow whose execution state has changed.

The body of each publication is an XML message giving 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:

```
<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 notified for both brokers and message flows are Started and Stopped.

These events are nonpersistent, retained publications.

Subscriptions and topics

Events are published to provide notification of changes to the subscription tables, changes in the list of defined topics or their access control lists.

Registration or deregistration of subscriptions causes events to be published using the following system topics:

```
$SYS/Broker/broker_name/Subscription/topic
```

where:

broker_name
is the name of the broker issuing this message.

topic is the original topic on which the subscription is being, or was, registered.

The body of each publication is an XML message giving additional information concerning the registration or deregistration request.

These events are nonpersistent, non-retained publications.

Examples:

Here are example event messages for a subscription being created, changed, and deleted.

Subscription created:

```
Event publication topic = "$SYS/Broker/Broker1/Subscription/stock/IBM"

<Broker uuid="1234" label="Broker1" version="1">
  <ControlGroup>
    <DynamicSubscriptionEngine>
      <Topic name="stock/IBM">
        <Create>
          <Subscription
            clientId="mqrfh2:Broker1:client1queue"
            subscriptionPoint="poundsSterling"
            filter="currentPrice>100"
            user="Fred"
            persistent="true"
            localOnly="false"
            pubOnReqOnly="false"
            informIfRet="true"
            expiryTimeStamp="2000-12-31 23:59:59"
            createTimeStamp="2000-01-01 00:00:00"
            tempDynamicQueue="false"
            clientContext="hex digits"/>
        </Create>
      </Topic>
    </DynamicSubscriptionEngine>
  </ControlGroup>
</Broker>
```

Subscription changed:

```
Event publication topic = "$SYS/Broker/Broker1/Subscription/stock/IBM"

<Broker uuid="1234" label="Broker1" version="1">
  <ControlGroup>
    <DynamicSubscriptionEngine>
      <Topic name="stock/IBM">
        <Change>
          <Subscription
            clientId="mqrfh2:Broker1:client1queue"
            subscriptionPoint="poundsSterling"
            filter="currentPrice>100"
            user="Fred"
            persistent="false"
            localOnly="true"
            pubOnReqOnly="true"
            informIfRet="false"
            expiryTimeStamp="2005-12-31 23:59:59"
            createTimeStamp="2000-01-01 00:00:00"
            tempDynamicQueue="false"
            clientContext="hex digits"/>
        </Change>
      </Topic>
    </DynamicSubscriptionEngine>
  </ControlGroup>
</Broker>
```

Subscription deleted (deregistered):

```
Event publication topic = "$SYS/Broker/Broker1/Subscription/stock/IBM"

<Broker uuid="1234" label="Broker1" version="1">
  <ControlGroup>
    <DynamicSubscriptionEngine>
      <Topic name="stock/IBM">
        <Delete>
          <Subscription
            clientId="mqrfh2:Broker1:client1queue"
            subscriptionPoint="poundsSterling"
            filter="currentPrice>100"
            user="Fred"/>
        </Delete>
      </Topic>
    </DynamicSubscriptionEngine>
  </ControlGroup>
</Broker>
```

Operational warnings

Operational warnings provide notification about expired subscriptions or retained publications.

See "Expired publications and subscriptions" for further information.

Expired publications and subscriptions

Notification of the expiry of retained publications and subscriptions is published on the following system topics:

`$SYS/Broker/`*`broker_name`*`/warning/expiry/Publication/`*`timestamp`*`/`*`topic`*

`$SYS/Broker/`*`broker_name`*`/warning/expiry/Subscription/`*`timestamp`*`/`*`topic`*

where:

broker_name
is the name of the broker issuing this message.

timestamp
is the timestamp of the expiry of the subscription or retained publication (expressed as a GMT time).

topic is the topic of the subscription or retained publication.

These events are nonpersistent, non-retained publications.

Examples:

Following are example event messages for a retained publication and subscription having expired.

Publication expired:

```
Event publication topic =
"$SYS/Broker/Broker1/warning/expiry/Publication/2000-12-31
23:59:59/stock/IBM"

<Broker uuid="1234" label="Broker1" version="1">
  <ControlGroup>
    <DynamicSubscriptionEngine>
      <Topic name="stock/IBM">
        <Delete>
          <RetainedPublication
            subscriptionPoint="poundsSterling"/>
        </Delete>
      </Topic>
    </DynamicSubscriptionEngine>
  </ControlGroup>
</Broker>
```

Subscription expired:

```
Event publication topic =
"$SYS/Broker/Broker1/warning/expiry/Subscription/2000-12-31
23:59:59/stock/IBM"

<Broker uuid="1234" label="Broker1" version="1">
  <ControlGroup>
    <DynamicSubscriptionEngine>
      <Topic name="stock/IBM">
        <Delete>
          <Subscription
            clientId="mqrfh2:Broker1:client1queue"
            subscriptionPoint="poundsSterling"
            filter="currentPrice>100"
            user="Fred"/>
        </Delete>
      </Topic>
    </DynamicSubscriptionEngine>
  </ControlGroup>
</Broker>
```

Notification message schema

The following specification describes the structure of all valid notification messages. This example describes the structure of the messages only. It says nothing 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>
. <ControlGroup>
. . <DynamicSubscriptionEngine>
. . . <Create>
. . . . <Neighbor neighbor_identifier neighbor_attributes>
. . . . . <MQBrokerConnection mqbrokerconnection_attributes>
. . . . <Neighbor neighbor_identifier neighbor_attributes>
. . . . <Topic topic_identifier>
. . . . . <ACLEntry>
. . . . . <Subscription>
. . . <Change>
. . . . <Neighbor neighbor_identifier neighbor_attributes>
. . . <Delete>
. . . . <AllNeighbors>
. . . . <Neighbor neighbor_identifier>
. . . . <Topic topic_identifier>
. . . . <AllACLEntries>
. . . . <AllSubscriptions>
. . . . <AllRetainedPublications>
. . . <Topic topic_identifier>
. . . . <Create>
. . . . . <ACLEntry acl_identifier acl_attributes>
. . . . . <Subscription subscription_identifier subscription_attributes>
. . . . <Change>
. . . . . <ACLEntry acl_identifier acl_attributes>
. . . . . <Subscription subscription_identifier subscription_attributes>
. . . . <Delete>
. . . . . <AllACLEntries>
. . . . . <ACLEntry acl_identifier>
. . . . . <AllSubscriptions>
. . . . . <Subscription subscription_identifier>
. . . . . <RetainedPublication retained_publication_identifier>
```

Identifiers and attributes used in the schema

Items shown as `[ .... ]` are optional attributes. Items shown as {`xxx` | `yyy`} mean that the value can be one of the alternatives given. Items shown in *`italics`* mean that the variable can have any value.

`uuid` The universally unique identifier for WebSphere Message Broker objects (for example, nodes in a message flow)

`client identifier`
The delivery destination for the subscriber. For example:

```
mqrfh2:WBRK_QM:subscriberqueue
mqrfh:WBRK_QM:subscriberqueue2:CorrelId
```

`identifier`
`uuid="`*`uuid`*`"`

`label` `label="`*`label`*`"`

`message_flow_identifier`
`uuid="`*`message_flow_identifier`*`"` //This is usually a uuid

message_flow_attributes

```
[label='label']
[additionalInstances="number"]
[commitCount="number"]
[commitInterval"number"]
[coordinatedTransaction={"yes" | "no" }]
```

node_identifier

uuid='*node_flow_identifier*' //This is usually a number of concatenated uuids

node_attributes

```
[label='label']
```

Others vary according to type. All are optional. See the node descriptions for details.

connection_identifier

```
sourceNode='source_node identifier'
sourceTerminal='source_terminal_name'
targetNode='target_node_identifier'
targetTerminal='target_terminal_name'
```

neighbor_identifier

```
name="uuid"
```

neighbor_attributes

```
collectiveId={"" | "uuid"}
```

CollectiveId is "" if the neighbor is in the same collective as the broker; otherwise, it is the identifier for the neighbor's collective.

mqbrokerconnection_attributes

```
queueManagerName="queueManagerName"
```

topic_identifier

```
name="topicName"
```

acl_identifier

```
principalName="userIdentifier"
```

acl_attributes

```
principalType={"user" | "group"}
[publish={"yes" | "no" | "inherit"}]
[subscribe={"yes" | "no" | "inherit"}]
[persistent={"yes" | "no" | "inherit"}]
```

subscription_identifier

```
clientId="client identifier"
[subscriptionPoint="subscriptionPointName"]
[filter="filterExpression"]
```

subscription_attributes

```
userId="userIdentifier"
persistent={"true" | "false" | "asPublish" | "asQDef"}
localOnly={"true" | "false"}
pubOnReqOnly={"true" | "false"}
informIfRet={"true" | "false"}
expiryTimeStamp={"GMTTimeStamp" | "0" }
createTimeStamp="GMTTimeStamp"
tempDynamicQueue={"true" | "false"}
clientContext="clientContext"
```

retainedpublication_identifier

```
[subscriptionPoint="subscriptionPointName"]
```

WebSphere MQ facilities

WebSphere Message Broker 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 very useful in diagnosing errors that occur in your broker domain. For example, if the Configuration Manager cannot communicate with a broker, the channels that connect them might be wrongly configured, or experiencing network problems.

On the distributed platforms, 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's subdirectory.

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 platforms, records are written to the /var/mqm/errors directory. WebSphere MQ writes one line for each FFST to the syslog containing the name of the FFST file, 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 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 WebSphere Message Broker broker domain.

Database facilities

The database products used by WebSphere Message Broker also record information that might be useful if you have any problems with their access. Refer to the database product documentation for details of logs and other problem determination options.

Database logs

DB2 has a number of facilities that assist you with problem diagnosis and recovery. For example, there are the error logs, db2diag.log on distributed platforms, and db2alert.log on z/OS, that contain error and alert information recorded by various components of the DB2 product. Refer to the *DB2 Troubleshooting Guide* for comprehensive information on what options are available, how to use them, and how to interpret the information provided.

Other database products have similar logs; consult your database administrator or database documentation for more information.

ODBC trace

You can trace ODBC activity; follow the instructions in "ODBC trace" on page 118.

Trace information is output to the location specified in the .odbc.ini file, or in the ODBC configuration on Windows. By default, this is /var/wmqi/odbc/odbctrace.out on non-Windows.

Other sources of diagnostic information on z/OS

This section describes some other files that you might find useful for problem determination on z/OS.

Files from a broker, Configuration Manager, or User Name Server

A broker, Configuration Manager, and User Name Server puts files into the HFS. If you do not know where the files are for the broker, look at the started task procedure or job output and locate COHFS. This is set to a value like /u/argo/VCP0BRK. Be careful to access the HFS on the correct system. If the HFS is not shared, there could be the same directory on each MVS image. To be sure, logon to the system where the broker is running and access the file system. You can use the TSO ISHell command, or go into OMVS itself. Go to the directory specified and select the output directory. The following files should appear in this directory:

- stderr
- stdout
- stderr.bipcvp
- stdout.bipcvp
- traceodbc

When submitting JCL from the component PDSE, output files are written to the ++HOME++ location. The following files should appear in this directory:

- ENVFILE
- bipprof
- *.err
- *.out

Files in the home directory

Other files are stored in the home directory of the user ID of the started task. The MVS command D A, jobname displays the user ID of the job. The directory of the user ID is typically /u/userid but you should check this with your system programmer, or issue the TSO command LU userid OMVS. If you are authorized, this command displays text including the HOME statement.

There might be several different sorts of files in the home directory, such as core dump files.

Problems accessing files in the HFS

If you cannot access an HFS file or directory, it might be that you do not have permission to do so. This might mean that you are not able to issue certain commands. You need to ask the owner of the file or directory to give you permission to use the file.

Solutions to similar problems

There are a variety of resources that contain information about problems with IBM licensed programs and their use. Searching these resources might help you to find information that helps solve your problem.

Useful Web sites

There are a variety of Web sites 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:

- WebSphere MQ family 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, and they regularly search the database to see if problems 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 already 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).

Part 3. Appendixes

Appendix. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this information in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this information. The furnishing of this information does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032,
Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION ″AS IS″ WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the information. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this information at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM United Kingdom Laboratories,
Mail Point 151,
Hursley Park,
Winchester,
Hampshire,
England
SO21 2JN

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Programming License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information includes examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

(C) (*your company name*) (*year*). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _*enter the year or years*_. All rights reserved.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

AIX
CICS
Cloudscape
DB2
DB2 Connect
DB2 Universal Database
developerWorks
Domino
Everyplace
FFST
First Failure Support Technology
IBM
IBMLink
IMS
IMS/ESA
iSeries
Language Environment
Lotus
MQSeries
MVS
NetView
OS/400
OS/390
pSeries
RACF
Rational
Redbooks
RETAIN
RS/6000
SupportPac
Tivoli
VisualAge
WebSphere
xSeries
z/OS
zSeries

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel and Pentium are trademarks or registered trademark of Intel Corporation or its subsidiaries in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Index

Printed in USA