10.0

IBM MQ Reference

Note

Before using this information and the product it supports, read the information in “Notices” on page 1455.

This edition applies to version 9 release 4 of IBM® MQ and to all subsequent releases and modifications until otherwise indicated in new editions.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 2007, 2026.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Reference **5**
- Configuration reference PDF 5
- Administration reference PDF 5
- Developing applications reference PDF 5
- IBM MQ Telemetry Reference 5
  - MQ Telemetry Transport format and protocol 5
  - MQXR properties 6
  - AuthCallback MQXR class 6
- Security reference 7
  - The API exit 8
  - The API-crossing exit 9
  - Certificate validation and trust policy design on AIX, Linux, and Windows systems 10
  - Managed File Transfer security reference 23
  - Cryptographic hardware 41
  - IBM MQ rules for SSLPEER values 42
  - GSKit: Digital certificate signature algorithms compliant with FIPS 140-3 43
  - GSKit return codes used in AMS messages 44
- Monitoring reference 50
  - Structure data types 50
  - Object attributes for event data 74
  - Authority Record attributes 119
  - Event message reference 119
- Messages and reason codes 244
  - AMQ messages on Multiplatforms 244
  - IBM MQ Console messages 246
  - IBM MQ Internet Pass-Thru messages 247
  - JSON format diagnostic messages 285
  - MFT messages and return codes 287
  - REST API messages 287
  - Telemetry messages 288
  - IBM MQ for z/OS messages, completion, and reason codes 289
  - API completion and reason codes 1151
  - PCF reason codes 1360
  - Token authentication error codes 1432
  - HTTP error codes 1440
  - Transport Layer Security (TLS) return codes 1441
  - WCF custom channel exceptions 1446

Notices **1455**
- Programming interface information 1456
- Trademarks 1456

IBM MQ Reference

Use the reference information in this section to accomplish the tasks that address your business needs.

Configuration reference PDF

From May 2021, the configuration reference information has moved to a separate PDF.

`mq93.refconfig.pdf` is available in the same directory as `mq93.reference.pdf`.

Administration reference PDF

From May 2021, the administration reference information has moved to a separate PDF.

`mq93.refadmin.pdf` is available in the same directory as `mq93.reference.pdf`.

Developing applications reference PDF

From May 2021, the developing applications reference information has moved to a separate PDF.

`mq93.refdev.pdf` is available in the same directory as `mq93.reference.pdf`.

ALW IBM MQ Telemetry Reference

Information about the MQTT format and protocol, the MQXR properties, and the AuthCallback MQXR class.

ALW MQ Telemetry Transport format and protocol

MQ Telemetry Transport (MQTT) is a lightweight publish/subscribe protocol flowing over TCP/IP to connect large numbers of remote sensors and control devices. MQTT is used by specialized applications on small footprint devices that must tolerate low bandwidth and unreliable communication. You can use MQTT client apps from a business partner, or write your own client apps to use the published protocols. You can get sample client apps and supporting libraries from the Eclipse Paho project.

IBM MQ Telemetry in IBM MQ supports client apps that use the MQTT protocol. There are currently two specifications of this protocol:

- The MQTT 3.1.1 Oasis standard
- The MQTT.org Protocol Specification from `mqtt.org`

The Oasis standard is more recent. The functionality is almost identical to the `mqtt.org` specification. The MQTT 3.1.1 Oasis standard is supported in IBM MQ 8.0.0 Fix Pack 3 and later versions.

If your MQTT client comes from a source other than the Eclipse Paho project, or the (no longer available) IBM Messaging Telemetry Clients SupportPac, check the version of the MQTT protocol supported by the client. If your client supports a different level of the MQTT protocol, and does not work correctly with the IBM MQ Telemetry service, a thin conversion layer is required. Check with the source of your client to see if the conversion layer is available as an update to the client you intend to use.

© Copyright IBM Corp. 2007, 2026

ALW MQXR properties

MQXR property settings are stored in a platform-specific properties file: `mqxr_win.properties` or `mqxr_unix.properties`. You normally configure these properties by using MQSC admin commands or IBM MQ Explorer.

When you start a queue manager for the first time, the template version of the MQXR properties file for your platform is copied from the *mqinstall*`/mqxr/config` directory to the *mqinstall*`/qmgrs/`*qmgr_name*`/mqxr/config` directory.

You do not normally need to edit the MQXR properties file directly, because all properties except one can be configured through MQSC admin commands or IBM MQ Explorer. If you do decide to edit the file directly, stop the queue manager before you make your changes.

The property that you can only set by editing the file directly is **webcontentpath**. If your telemetry client app is a web app, you also need to serve the web app executable JavaScript to the browser. This requirement is explained in The MQTT messaging client for JavaScript(tm) and web apps. You use the **webcontentpath** property to specify the directory from which the web app executable files are served:

- By default, **webcontentpath** is not present in the MQXR properties file. If **webcontentpath** is not present, the MQ telemetry server serves the web app executable files from the following default location: *mqinstall*`/qmgrs/`*qmgr_name*`/mqxr/WebContent/`*your_client_app*
- if **webcontentpath** specifies a path, the MQ telemetry server serves the web app executable files from that location.
- if **webcontentpath** is present and blank, the MQ telemetry server does not serve the web app executable files.

Related concepts

Telemetry (MQXR) service

Related reference

"AuthCallback MQXR class" on page 6

`AuthCallback` is the sole class in package `com.ibm.mq.mqxr`. It specifies the interface definition that a telemetry server administrator needs when they write an `AuthCallback` in the MQXR server.

ALW AuthCallback MQXR class

`AuthCallback` is the sole class in package `com.ibm.mq.mqxr`. It specifies the interface definition that a telemetry server administrator needs when they write an `AuthCallback` in the MQXR server.

Class AuthCallback

```
java.lang.Object
    |
    └ com.ibm.mq.mqxr.AuthCallback
```

Implemented interface:

```
javax.security.auth.callback.Callback
```

```
public class AuthCallback
extends java.lang.Object
implements javax.security.auth.callback.Callback
```

Allows a JAAS login module (javax.security.auth.spi.LoginModule) to access IBM MQ Server objects.

Methods

getSSLSession

```
public javax.net.ssl.SSLSession getSSLSession()
```

Returns the `javax.net.ssl.SSLSession` associated with the client connection, or null if the client is connected using a plain text connection.

setSSLSession

```
public void setSSLSession(javax.net.ssl.SSLSession sslSession)
```

The **`sslSession`** parameter is set by the server to be the `sslSession` associated with the client connection, or null if the client is connected using a plain text connection.

Constructors

```
public AuthCallback()
```

Related concepts

Telemetry (MQXR) service

Telemetry channel JAAS configuration

Related tasks

Resolving problem: JAAS login module not called by the IBM MQ Telemetry service

Related reference

"MQXR properties" on page 6

MQXR property settings are stored in a platform-specific properties file: `mqxr_win.properties` or `mqxr_unix.properties`. You normally configure these properties by using MQSC admin commands or IBM MQ Explorer.

Security reference

Use the reference information in this section to help you configure security for IBM MQ.

Related concepts

"The API exit" on page 8

An *API exit* is a program module that monitors or modifies the function of MQI calls. An API exit comprises multiple *API exit functions*, each with its own entry point in the module.

"The API-crossing exit" on page 9

An *API-crossing exit* is a program that monitors or modifies the function of MQI calls issued by CICS applications on z/OS.

"Certificate validation and trust policy design on AIX, Linux, and Windows systems" on page 10

IBM MQ validates TLS certificates according to three types of policy, basic, standard, and none. Standard policy checking conforms to RFC 5280.

"Cryptographic hardware" on page 41

The way in which IBM MQ provides support for cryptographic hardware depends on which platform you are using.

"IBM MQ rules for SSLPEER values" on page 42

The **SSLPEER** attribute is used to check the Distinguished Name (DN) of the certificate from the peer queue manager or client at the other end of an IBM MQ channel. IBM MQ uses certain rules when comparing these values

CipherSpec mismatches

Authentication failures

Overview of Advanced Message Security interception on message channels

Related reference

"GSKit: Digital certificate signature algorithms compliant with FIPS 140-3 " on page 43

The list of digital certificate signature algorithms in IBM Global Security Kit (GSKit) that are compliant with FIPS 140-3

The API exit

An *API exit* is a program module that monitors or modifies the function of MQI calls. An API exit comprises multiple *API exit functions*, each with its own entry point in the module.

Note: The information in this section does not apply to IBM MQ for z/OS.

There are two categories of exit function:

An exit function that is associated with an MQI call

There are two exit functions in this category for each MQI call and an additional one for an MQGET call with the MQGMO_CONVERT option. The MQCONN and MQCONNX calls share the same exit functions.

For each MQI call, one of the two exit functions is invoked before the queue manager starts to process the call and the other is invoked after the queue manager has completed processing the call. The exit function for an MQGET call with the MQGMO_CONVERT option is invoked during the MQGET call, after the message has been retrieved from the queue by the queue manager but before any data conversion takes place. This allows, for example, a message to be decrypted before data conversion.

An exit function can inspect and modify any of the parameters on an MQI call. On an MQPUT call, for example, an exit function that is invoked before the processing of the call has started can:

- Inspect and modify the contents of the application data in the message being put
- Change the length of the application data in the message
- Modify the contents of the fields in the message descriptor structure, MQMD
- Modify the contents of the fields in the put message options structure, MQPMO

An exit function that is invoked before the processing of an MQI call has started can suppress the call completely. The exit function for an MQGET call with the MQGMO_CONVERT option can suppress data conversion of the message being retrieved.

Initialization and termination exit functions

There are two exit functions in this category, the initialization exit function and the termination exit function.

The initialization exit function is invoked by the queue manager when an application connects to the queue manager. Its primary purpose is to register exit functions and their entry points with the queue manager and perform any initialization processing. You do not have to register all the exit functions, only those that are required for this connection. When the application disconnects from the queue manager, the registrations are removed automatically.

The initialization exit function can also be used to acquire any storage required by the exit and examine the values of any environment variables.

The termination exit function is invoked by the queue manager when an application disconnects from the queue manager. Its purpose is to release any storage used by the exit and perform any required cleanup operations.

An API exit can issue calls to the MQI but, if it does, the API exit is not invoked recursively a second time. The following exit functions, however, are not able to issue MQI calls because the correct environment is not present at the time the exit functions are invoked:

- The initialization exit function
- The exit function for an MQCONN and MQCONNX call that is invoked *before* the queue manager starts to process the call
- The exit function for the MQDISC call that is invoked *after* the queue manager has completed processing the call
- The termination exit function

An API exit can also use other APIs that might be available; for example, it can issue calls to Db2®.

An API exit can be used with an IBM MQ client application, but it is important to note that the exit is invoked at the *server* end of an MQI channel. For more information, see Comparing link level security and application level security.

An API exit is written using the C programming language.

To enable an API exit, you must configure it. On IBM MQ for Multiplatforms, you do this by editing the IBM MQ configuration file, mqs.ini, and the queue manager configuration file, qm.ini, for each queue manager.

For a client, modify the `ApiExitLocal` stanza in the mqclient.ini file to identify API exit routines for a queue manager.

You configure an API exit by providing the following information:

- The descriptive name of the API exit.
- The name of the module and its location; for example, the full path name.
- The name of the entry point for the initialization exit function.
- The sequence in which the API exit is invoked relative to other API exits. You can configure more than one API exit for a queue manager.
- Optionally, any data to be passed to the API exit.

Notes:

- An API exit is not called for puts to:
  1. A streaming queue. Any API exit modified message put to the original queue, is also put to any associated streaming queue.
  2. The destination queue for a subscription. Any API exit modified message publication to the original topic, is put to the appropriate subscription destinations.
- For puts to distribution lists, the API exits are called only once before and after putting the message to the list of queues.

For more information about how to configure an API exit, see Configuring API exits.

For information about how to write an API exit, see Using and writing API exits.

z/OS The API-crossing exit

An *API-crossing exit* is a program that monitors or modifies the function of MQI calls issued by CICS applications on z/OS.

Note: The information in this section applies only to CICS applications on z/OS.

The API-crossing exit program is invoked by the CICS adapter and runs in the CICS address space.

The API-crossing exit is invoked for the following MQI calls only:

MQBUFMH
MQCB
MQCB_FUNCTION
MQCLOSE
MQCRTMH
MQCTL
MQDLTMH
MQGET
MQINQ
MQOPEN
MQPUT
MQPUT1
MQSET

MQSTAT
MQSUB
MQSUBRQ

For each MQI call, it is invoked once before the processing of the call has started and once after the processing of the call has been completed.

The exit program can determine the name of an MQI call and can inspect and modify any of the parameters on the call. If it is invoked before an MQI call is processed, it can suppress the call completely.

The exit program can use any of the APIs that a CICS task-related user exit can use; for example, the IMS, Db2, and CICS APIs. It can also use any of the MQI calls except MQCONN, MQCONNX, and MQDISC. However, any MQI calls issued by the exit program do not invoke the exit program a second time.

You can write an API-crossing exit in any programming language supported by IBM MQ for z/OS.

Before an API-crossing exit can be used, the exit program load module must be available when the CICS adapter connects to a queue manager. The load module is a CICS program that must be named CSQCAPX and reside in a library in the DFHRPL concatenation sequence. CSQCAPX must be defined in the CICS system definition file (CSD), and the program must be enabled.

An API-crossing exit can be managed using the CICS adapter control panels, CKQC. When CSQCAPX is loaded, a confirmation message is written to the adapter control panels or to the system console. The adapter control panels can also be used to enable or disable the exit program.

For more information about how to write and implement an API-crossing exit, see "The CICS-IBM MQ Adapter" section in the CICS Transaction Server for z/OS 4.1 product documentation. See CICS Transaction Server for z/OS 4.1, The CICS-IBM MQ adapter.

ALW Certificate validation and trust policy design on AIX, Linux, and Windows systems

IBM MQ validates TLS certificates according to three types of policy, basic, standard, and none. Standard policy checking conforms to RFC 5280.

The information in these topics applies to the following systems:

- IBM MQ for AIX® or Linux® systems
- IBM MQ for Windows systems

The following terms are used in this section:

Certificate policy
Determines which fields in a certificate are understood and processed.

OCSP policy
Determines which fields in an OCSP request or response are understood and processed.

CRL policy
Determines which fields in a certificate revocation list are understood and processed.

Path validation policy
Determines how the certificate, OCSP, and CRL policy types interact with each other to determine whether a certificate chain (a trust point "RootCA" to an end-entry "EE") is valid.

The basic and standard path validation policies are described separately because it reflects the implementation within IBM MQ for UNIX, Linux and Windows systems. However, the standard OCSP and CRL policies are the same as the basic policies, and the standard certificate policy is an extended version of the basic policy, so these policies are not described separately.

By default, IBM MQ applies basic policy validation first. If basic policy validation fails, IBM MQ applies standard policy (RFC 5280) validation. If basic policy validation succeeds, standard policy validation is not applied. Thus, a validation failure means that both basic and standard policy validation failed, possibly for different reasons. A validation success means that either basic policy validation succeeded and standard

policy validation was therefore not applied, or basic policy validation failed and standard policy validation succeeded.

Enforcing strict RFC 5280 compliance

To enforce strict RFC 5280 compliance, use the certificate validation policy configuration setting. This setting allows you to disable the basic policy, so that only the standard RFC 5280 policy is used. For more information about the certificate validation policy configuration setting, see Certificate validation policies in IBM MQ.

The following examples are digital certificates that are accepted by the basic certificate validation policy, but which are rejected by the RFC 5280 compliant standard policy. In order for a digital certificate chain to be trusted, the entire chain must satisfy the configured validation policy.

To view the full details of a digital certificate, use the **runmqakm** command:

```
runmqakm -cert -details -db key.kdb -pw password -label certificate_label
```

A certificate that has trust status enabled in the **runmqakm** output is not necessarily trusted for use in a TLS handshake. Trust status enabled means that the certificate is eligible to be used as a CA certificate to verify other certificates, if the certificate also satisfies the rules of the certificate validation policy. For more information about the RFC 5280 compliant standard certificate validation policy, see “Standard path validation policy” on page 19.

Example certificate 1 - incorrect key usage

This example shows a certificate where the key usage field does not comply with the standard certificate validation policy rules for a CA certificate. One of the requirements for a certificate to be valid for use as a CA certificate is that the key usage field must indicate that it is permitted to sign other certificates using the keyCertSign flag. A certificate without this flag cannot be used as a CA certificate.

```
Label : root
Key Size : 1024
Version : X509 V3
Serial : 54cb6f740c7ee410
Issuer : CN=Example Root CA,O=Example,C=GB
Subject : CN=Example Root CA,O=Example,C=GB
Not Before : 9 February 2012 17:19:00 GMT
Not After : 1 October 2019 18:19:00 GMT+01:00
Public Key
    30 81 9F 30 0D 06 09 2A 86 48 86 F7 0D 01 01 01
    05 00 03 81 8D 00 30 81 89 02 81 81 00 CC 44 D9
    25 6D 26 1C 9D B9 FF DE B8 AC 44 AB E3 64 80 44
    AF BE E0 00 93 53 92 33 F8 7E BD D7 71 ED 21 52
    24 75 DF D6 EE 3C 54 97 84 29 EA 93 4C 4A D1 19
    5D C1 A0 82 F5 74 E1 AD D9 87 10 D5 6A 2B 6F 90
    04 0F 7E 6E 85 6D 32 99 33 9C D9 BB 57 86 DE 68
    23 C9 F2 6D 53 E3 F5 FF D1 0B E7 23 19 3A F6 70
    6B C8 C7 EB DB 78 8E 8C 9E 55 58 66 B6 31 DB 40
    5F 6A 97 AB 12 D7 E2 3E 2E 79 EE 78 7B 02 03 01
    00 01
Public Key Type : RSA (1.2.840.113549.1.1.1)
Fingerprint : SHA1 :
    EE 68 D4 4F 73 4F F4 21 DE 1A 01 11 5E DE B1 B8
    DF 40 AA D8
Fingerprint : MD5 :
    50 B5 E9 B2 D7 35 05 6A DC 6D 4B 1E B2 F2 DF A4
Fingerprint : SHA256 :
    B4 D7 6E C4 47 26 24 C7 4F 41 C3 83 03 6F 5C C7
    07 11 61 E0 0E 36 59 1F 1C E6 69 39 2D 18 05 D2
Extensions
    basicConstraints
        ca = true
        pathLen = 1239876
        critical
    key usage: encipherOnly
Signature Algorithm : SHA256WithRSASignature (1.2.840.113549.1.1.11)
Value
    9D AE 54 A9 9D 68 01 68 15 B5 53 9F 96 C9 5B D1
    52 40 DB CB 33 AF FD B9 26 D5 90 3F 1E 0B FC A6
    D9 8C 04 90 EB AA FD A8 7A 3C AB 60 5F 20 4F 0D
    7B 73 41 27 6A 2B BF 8C 99 91 B6 49 96 82 6A 24
```

```
    0A E8 B9 A5 AF 69 3D 2C A3 3C C8 12 39 FB 56 58
    4E 2A FE AC AC 10 89 53 B1 8F 0F C0 50 BF 5E 00
    91 64 B4 A1 4C 9A 4E D5 1F 38 7C AD 32 A9 8A E1
    91 16 2C 6D 1E 4A CA 99 8D CC 22 CD BF 90 49 FC
Trust Status : Enabled
```

In this example, the key usage field contains only the encipherOnly flag. The keyCertSign flag is not set, so this certificate is not permitted to sign other certificates. It therefore cannot be used as a CA certificate.

Example certificate 2 - missing basic constraints extension

This example shows a certificate which lacks the basic constraints extension. The basic constraints extension is used to indicate whether this certificate is permitted for use as a CA. It is also used to indicate the maximum length of any certificate chain which can be signed by the certificate. The standard certificate validation policy requires that the certificate has a basic constraints extension with the isCA flag set in order to be used as a CA.

```
Label : root
Key Size : 1024
Version : X509 V3
Serial : 1c7dfea316570bf6
Issuer : CN=Second Example Root CA,O=Example,C=GB
Subject : CN=Second Example Root CA,O=Example,C=GB
Not Before : 9 February 2012 17:18:22 GMT
Not After : 1 October 2019 18:18:22 GMT+01:00
Public Key
    30 81 9F 30 0D 06 09 2A 86 48 86 F7 0D 01 01 01
    05 00 03 81 8D 00 30 81 89 02 81 81 00 B2 70 49
    7C AE 1B A7 B3 06 49 6C 99 19 BC A8 77 BE 86 33
    21 6B C9 26 CC A6 28 52 9F 7B CF 03 A4 37 A7 4D
    6B 06 AA ED 7D 58 E3 70 F3 F7 C1 06 DA E8 27 C6
    3D 1B AC FA EF AA 59 7A 9A AB C1 14 4E AF 13 14
    4B 71 CA 8D FE C3 F5 2F E8 AC AD EF 21 80 6D 12
    89 4A 2A 84 AA 9D E0 4F C1 93 B1 3E 16 E8 3C 75
    39 2A 74 1E 90 CC B1 C3 2B 1D 55 26 76 D2 65 C1
    06 47 2A BF 79 96 42 76 A9 6E 65 88 5F 02 03 01
    00 01
Public Key Type : RSA (1.2.840.113549.1.1.1)
Fingerprint : SHA1 :
    33 9F A1 81 43 F1 43 95 48 A5 66 B4 CD 98 E8 15
    9C B3 CA 90
Fingerprint : MD5 :
    91 EA D9 C0 2C 05 5B E2 CD 0B F6 DD 8A 11 44 23
Fingerprint : SHA256 :
    62 46 35 0B 0E A1 A7 2A D5 74 70 0F AA 47 9A 9C
    6B 80 1B F1 0B 4C 81 05 85 0E 91 11 A4 21 D2 34
Extensions
    key usage: digitalSignature, keyCertSign
Signature Algorithm : SHA256WithRSASignature (1.2.840.113549.1.1.11)
Value
    79 34 BA 5B 6F DC 06 A3 99 24 4E 8A 2B 27 05 47
    0D 4D BE 6A 77 D1 1D 5F 54 82 9D CC F6 92 D4 9A
    AB 4D B6 DD 6E AD 86 C3 6A A3 32 E3 B3 ED E0 62
    4A EB 51 08 AC BE 49 9E 9C D7 FE AE C8 9D 17 16
    68 31 6B F4 BA 74 1E 4F 5F 05 48 9F E7 46 BA DC
    17 7A 60 88 F8 5B DB 3C 51 D4 98 97 28 82 CF 36
    47 DA D2 0F 47 FF 70 EA 45 3A 49 66 E6 E2 F9 67
    2C C8 3E 24 A2 3B EC 76 1F D6 31 2B BD A9 B5 08
Trust Status : Enabled
```

In this example, the certificate lacks the basic constraints field entirely. Therefore this certificate cannot be used as a CA certificate.

Example certificate 3 - intermediate CA with old version of X.509

This example shows an intermediate CA certificate which is at X.509 version 1. The standard certificate validation policy requires that all intermediate CA certificates must be at least X.509 version 3. Root CA certificates are exempt from this requirement as there are still some commonly used version 1 root CA certificates in existence. However, this exemption might change in future.

```
Label : intermediate
Key Size : 1024
Version : X509 V1
Serial : 02
Issuer : CN=Test Root CA,O=Example,C=GB
Subject : CN=Test Intermediate CA,O=Example,C=GB
```

```
Not Before : 10 February 2012 17:33:45 GMT
Not After : 11 April 2018 18:33:45 GMT+01:00
Public Key
    30 81 9F 30 0D 06 09 2A 86 48 86 F7 0D 01 01 01
    05 00 03 81 8D 00 30 81 89 02 81 81 00 C0 07 C2
    D0 9F 84 DB 7C 20 8F 51 F9 C2 1A 3F CF E2 D7 F2
    F1 56 F2 A4 8F 8F 06 B7 3B 01 31 DE 7C CC 03 63
    AA D3 2F 1C 50 15 E3 56 80 40 7D FF 75 87 D3 F3
    00 89 9A 26 F5 57 05 FA 4F ED 3B DD 93 FA F2 DF
    38 26 D4 3A 92 51 CC F3 70 27 42 7A 9F AD 51 45
    67 B7 AE 11 AD 4F 2D AB D2 CF 73 E6 F0 45 92 F0
    47 16 66 7E 01 C7 76 A3 7B EC D2 76 3F E5 15 EC
    D7 72 2C FE 14 F5 78 83 AA C4 20 AB F7 02 03 01
    00 01
Public Key Type : RSA (1.2.840.113549.1.1.1)
Fingerprint : SHA1 :
    DE BB 75 4B 14 E1 44 B9 B6 44 33 97 49 D0 82 6D
    81 F2 2F DE
Fingerprint : MD5 :
    72 49 44 42 E2 E6 89 F1 CC 37 C9 F6 B5 8F F3 AE
Fingerprint : SHA256 :
    83 A4 52 AF 49 34 F1 DC 49 E6 95 AE 93 67 80 13
    C2 64 D9 26 22 A0 E8 0A 5A A9 71 EC E8 33 E1 D1
Signature Algorithm : SHA256WithRSASignature (1.2.840.113549.1.1.11)
Value
    40 4A 09 94 A0 18 07 5E 96 D7 A6 52 6B 8D 20 50
    E8 91 F7 7E EA 76 B4 08 DF 76 66 1F FA FF 91 79
    2E E0 66 8B 9F 40 FA 14 13 79 81 DB 31 A5 55 1D
    44 67 41 F4 EA 1A F7 83 4F 21 F4 43 78 4E F8 5E
    6F B2 B8 3A F7 6B B4 F5 C6 F8 EB 4C BF 62 6F 3E
    C7 20 EC 53 B3 40 51 36 C1 0A 4E 73 ED 74 D1 93
    02 C5 FB 61 F7 87 64 A5 94 06 7D 25 7C E3 73 DD
    08 D4 07 D0 A4 3F 77 88 12 59 DB A4 DB 68 8F C1
Trust Status : Enabled
```

In this example, the version field is X.509 V1. This certificate is an X.509 version 1 certificate and therefore cannot be used as an intermediate CA.

ALW Basic and standard certificate policies

The basic and standard certificate policies support the same fields: the standard policy supports additional certificate extensions.

The supported fields for both the basic and standard policies are as follows:

- OuterSigAlgID [1]
- Signature [2]
- Version
- SerialNumber
- InnerSigAlgID [3]
- Issuer
- Validity
- SubjectName
- SubjectPublicKeyInfo
- IssuerUniqueID
- SubjectUniqueID

The supported extensions for the basic policy are as follows. Where an entry is marked as "not supported", IBM MQ does not attempt to process extensions containing a field of that specific type, but does process other types of the same extension.

- AuthorityKeyID

[1] This field is called *signatureAlgorithm* in RFC 5280.
[2] This field is called *signatureValue* in RFC 5280.
[3] This field is called *signature* in RFC 5280.

- AuthorityInfoAccess
- SubjectKeyID
- IssuerAltName
- SubjectAltName
- KeyUsage
- BasicConstraints
- PrivateKeyUsage
- CRLDistributionPoints
  - DistributionPoint
    - DistributionPointName (X.500 Name and LDAP Format URI only)
    - NameRelativeToCRLIssuer (not supported)
    - Reasons (ignored)
    - CRLIssuer fields (not supported)

The supported extensions for the standard policy are all those listed for the basic policy and those in the following list. Where an entry is marked as "not supported", IBM MQ does not attempt to process extensions containing a field of that specific type, but does process other types of the same extension.

- NameConstraints
- ExtendedKeyUsage
- CertificatePolicies
  - PolicyInformation
    - PolicyIdentifier
    - PolicyQualifiers (not supported)
- PolicyMappings
- PolicyConstraints

ALW Basic and standard OCSP policies

The basic and standard OCSP policies support the same fields.

The supported fields for a request are as follows. Where an entry is marked as "not supported", IBM MQ does not attempt to process a request containing a field of that specific type, but does process other requests containing the same higher-level field.

- Signature (Optional)
- Version (Version 1 Only)
- RequesterName (Optional)
- RequestList (single request only)
  - CertID [4]
  - singleRequestExtensions (not supported)
- RequestExtensions
  - Nonce (if enabled)

The supported fields for a response are as follows:

- ResponseStatus
- Response

[4] This field is called reqCert in RFC 2560

- responseType (id-pkix-ocsp-basic)
- BasicOCSPResponse
  - Signature
  - Certs
    - Extensions
    - extendedKeyUsage
      - id-kp-OCSPSigning
    - id-pkix-ocsp-nocheck
  - ResponseData
    - Version (Version 1 Only)
    - ResponderID (by name or by hash)
    - ProducedAt (ignored)
    - Responses (multiple responses supported)
      - SingleResponse
        - certID
        - certStatus
          - RevokedInfo (ignored)
        - thisUpdate (ignored)
        - nextUpdate
        - singleExtensions (ignored)
    - responseExtensions
      - Nonce (if enabled)

ALW Basic and standard CRL policies

The basic and standard CRL policies support the same fields and extensions.

The supported fields for these policies are as follows:

- OuterSigAlgID [5]
- Signature [6]
- Version
- InnerSigAlgID [7]
- Issuer
- ThisUpdate
- NextUpdate
- RevokedCertificate
  - UserCertificate
  - RevocationDate

There are no supported CRLEntry extensions.

[5] This field is called *signatureAlgorithm* in RFC 5280.
[6] This field is called *signatureValue* in RFC 5280.
[7] This field is called *signature* in RFC 5280.

The supported CRL extensions for these policies are as follows. Where an entry is marked as "not supported", IBM MQ does not attempt to process extensions containing a field of that specific type, but does process other types of the same extension.

- AuthorityKeyID
- IssuerAltName
- CRLNumber
- IssuingDistributionPoint
  - DistributionPoint
  - DistributionPointName
    - FullName (X.500 Name and LDAP Format URI only)
    - NameRelativeToCRLIssuer (not supported)
  - Reasons (ignored)
  - CRLIssuer
  - OnlyContainsUserCerts (not supported)
  - OnlyContainsCACerts (not supported)
  - OnlySomeReasons (not supported)
  - IndirectCRL [8] (rejected)

ALW Basic path validation policy

The basic path validation policy determines how the certificate, OCSP, and CRL policy types interact with each other to determine if a certificate chain is valid.

The validation of a chain is performed in the following manner (but not necessarily in the following order):

1. Ensure that the name of the certificate's issuer is equal to the subject name in the previous certificate, and that there is not an empty issuer name in this certificate or the previous certificate subject name. If no previous certificate exists in the path and this is the first certificate in the chain, ensure that the issuer and subject name are identical and that the certificate is trusted.

   Note: IBM MQ for UNIX, Linux and Windows systems will fail path validation in situations where the previous certificate in a path has the same subject name as the current certificate.
2. Ensure that the signature algorithm used to actually sign the certificate matches the signature algorithm indicated within the certificate, by ensuring that the issuer signature algorithm identifier in the certificate matches the algorithm identifier in the signature data.
3. Ensure that the certificate was signed by the issuer, using the subject public key from the previous certificate in the path to verify the signature on the certificate. If no previous certificate exists and this is the first certificate, use the subject public key of the certificate to verify the signature on it. IBM MQ supports DSA and RSA signature algorithms; however it does not support DSA Parameter Inheritance.
4. Ensure that the certificate is a known X509 version, unique IDs are not present for version 1 certificates, and extensions are not present for version 1 and version 2 certificates.
5. Ensure that the certificate has not expired, or not been activated yet, and that its validity period is good [9] .
6. Ensure that there are no unknown critical extensions or any duplicate extensions.
7. Ensure that the certificate has not been revoked. Here, the following operations apply:

[8] IndirectCRL extensions will result in CRL validation failing. IndirectCRL extensions must not be used because they cause identified certificates to not be rejected.

[9] There are no checks to ensure the subject's validity is within bounds of the issuer's validity. This is not required, and it has been shown that certificates from some CAs do not pass such a check.

a. If the OCSP connection is enabled and a Responder Address is configured or the Certificate has a valid AuthorityInfoAccess extension specifying a HTTP format GENERALNAME_uniformResourceID check revocation status with OCSP.

b. If revocation status from “7.a” on page 17 above is undetermined the CRLDistributionPoints extension is checked for a list of X.500 distinguished name GENERALNAME_directoryname and URI GENERALNAME_uniformResourceID. Only LDAP, HTTP and FILE format URIs are supported. If the extension is not present, or use of the CRLDistributionPoints extension results in undetermined status and the extension is not Critical, the certificate's issuer's name is used to query revocation status. A CRL database (LDAP) is then queried for CRLs. If the certificate is not the last certificate, or if the last certificate has the basic constraint extension with the "isCA" flag turned on, the database is queried for ARLs and CRLs instead. If CRL checking is enabled, and no CRL database can be queried, the certificate is treated as revoked. Currently, the X500 directory name form and the LDAP/HTTP/FILE URI forms are the only supported name forms used to look up CRLs and ARLs [10] .

   Note: RelativeDistinguishedNames are not supported.

c. If revocation status from both “7.a” on page 17 and “7.b” on page 17 is undetermined, IBM MQ checks the *OCSPAuthentication* configuration setting to decide whether to allow the connection. [11]

8. If the issuerAltName extension is marked critical, ensure that the name forms are recognized. The following general name forms are currently recognized:
   - rfc822
   - DNS
   - directory
   - URI
   - IPAddress(v4/v6)
9. If the subjectAltName extension is marked critical, ensure that the name forms are recognized. The following general name forms are currently recognized:
   - rfc822
   - DNS
   - directory
   - URI
   - IPAddress(v4/v6)
10. If the KeyUsage extension is critical on a non-EE certificate, ensure that the keyCertSign flag is on, and ensure that if the BasicConstraints extension is present, the "isCA" flag is true.
11. If the BasicConstraints extension is present, the following checks are made:
    - If the "isCA" flag is false, ensure the certificate is the last certificate in the chain and that the pathLength field is not present.
    - If the "isCA" flag is true and the certificate is NOT the last certificate in the chain, ensure that the number of certificates until the last certificate in the chain is not greater than the pathLength field.
12. The AuthorityKeyID extension is not used for path validation, but is used when building the certificate chain.
13. The SubjectKeyID extension is not used for path validation, but is used when building the certificate chain.

---

[10] After they are retrieved from the database, ARLs are evaluated in exactly the same fashion as CRLs. Many CAs do not issue ARLs. However, IBM MQ will look for ARLs and CRLs if checking a CA certificate for revocation status.

[11] If *OCSPAuthentication* is set to WARN, IBM MQ logs the unknown revocation status and allows the connection to continue.

14. The PrivateKeyUsagePeriod extension is ignored by the validation engine, because it cannot determine when the CA actually signed the certificate. The extension is always non-critical and therefore can be safely ignored.

An OCSP Response is also validated to ensure that the response itself is valid. Validation is performed in the following manner (but not necessarily the following order):

1. Ensure that response status is `Successful` and the response type is PKIX_AD_OCSP_basic.r
2. Ensure that response version data is present and the response is the correct version (Version 1)
3. Ensure that the response is correctly signed. The signature will be rejected if the signer does not meet at least one of the following criteria:
   - The signer matches a local configuration of OCSP signing authority [12] for the certificate.
   - The signer is using the CA key for which the public key is contained in the CA certificate, that is, the CA itself is directly signing the response.
   - The signer is a direct sub-ordinate of the CA that signed the certificate for which revocation information is being checked and is authorized by the CA by including the value of id-ad-ocspSigning in an ExtendedKeyUsage extension.

     Note: Revocation checking of the response signer certificate is not performed if the id-pkix-ocsp-nocheck extension is present.
4. Ensure that response hash algorithm, serialNumber, issuerNameHash, and issuerKeyHash match those of the request.
5. Ensure that the response has not expired, that is, that the nextUpdate time is greater than the current time. [13]
6. Ensure that the certificate has valid revocation status.

The validation of a CRL is also performed to ensure that the CRL itself is valid, and is performed in the following manner (but not necessarily the following order):

1. Ensure that the signature algorithm used to actually sign the CRL matches the signature algorithm indicated within the CRL, by ensuring that the issuer signature algorithm identifier in the CRL matches the algorithm identifier in the signature data.
2. Ensure that the CRL was signed by the issuer of certificate in question, verifying that the CRL has been signed with the key of the certificate issuer.
3. Ensure that the CRL has not expired [14] , or not been activated yet, and that its validity period is good.
4. Ensure that if the version field is present, it is version 2. Otherwise the CRL is version 1 and must not have any extensions. However, IBM MQ for UNIX, Linux and Windows systems only verifies that no critical extensions are present for a version 1 CRL.
5. Ensure that the certificate in question is on the revokedCertificates field list and that the revocation date is not in the future.
6. Ensure that there are no duplicate extensions.
7. If unknown critical extensions, including critical entry extensions, are detected in the CRL, this causes identified certificates to be treated as revoked [15] (provided the CRL passes all other checks).

---

[12] This is a Certificate in the KeyStore a user has installed and that has Trust Status set.

[13] If no current OCSP responses are returned from the responder, IBM MQ will attempt to use out of date responses in determining the revocation status of a Certificate. IBM MQ attempts to use out of date Responses so that security will not be adversely reduced.

[14] If no current CRLs are found, IBM MQ for UNIX, Linux and Windows systems will attempt to use out of date CRLs to determine the revocation status of a Certificate. It is not clearly specified in RFC 5280 what action to take in the event of no current CRLs. IBM MQ for UNIX, Linux and Windows systems attempt to use out of date CRLs so that security will not be adversely reduced.

[15] ITU X.509 and RFC 5280 are in conflict in this case because the RFC mandates that CRLs with unknown critical extensions must fail validation. However, ITU X.509 requires that identified certificates must still

8. If the authorityKeyID extension in the CRL and the subjectKeyID in the CA certificate are present and if the keyIdentifier field is present within the authorityKeyID of the CRL, match it with the CACertificate's subjectKeyID.
9. If the issuerAltName extension is marked critical, ensure that the name forms are recognized. The following general name forms are currently recognized:
   - rfc822
   - DNS
   - directory
   - URI
   - IPAddress(v4/v6)
10. If the issuingDistributionPoint extension is present in the CRL, process as follows:
    - If the issuingDistributionPoint specifies an InDirectCRL then fail the CRL validation.
    - If the issuingDistributionPoint indicates that a CRLDistributionPoint is present but no DistributionPointName is found, fail the CRL validation
    - If the issuingDistributionPoint indicates that a CRLDistributionPoint is present and specifies a DistributionPointName ensure that it is a GeneralName or LDAP format URI that matches the name given by the certificate's CRLDistributionPoint or the certificate's issuer's name. If the DistributionPointName is not a GeneralName then the CRL validation will fail.

      Note: RelativeDistinguishedNames are not supported and will fail CRL validation if encountered.

ALW Standard path validation policy

The standard path validation policy determines how the certificate, OCSP, and CRL policy types interact with each other to determine if a certificate chain is valid. Standard policy checking conforms to RFC 5280.

Path validation uses the following concepts:

- A certification path of length *n*, where the trust point or root certificate is certificate 1, and the EE is *n*.
- A set of initial policy identifiers (each comprising a sequence of policy element identifiers), that identifies one or more certificate policies, any one of which is acceptable for the purposes of certification path processing, or the special value "any-policy". Currently this is always set to "any-policy".

  Note: IBM MQ for UNIX, Linux and Windows systems only supports policy identifiers that are created by IBM MQ for UNIX, Linux and Windows systems.
- Acceptable policy set: a set of certificate policy identifiers comprising the policy or policies recognized by the public key user, together with policies deemed equivalent through policy mapping. The initial value of the acceptable policy set is the special value "any-policy".
- Constrained subtrees: a set of root names defining a set of subtrees within which all subject names in subsequent certificates in the certification path can fall. The initial value is "unbounded".

---

be treated as revoked provided the CRL passes all other checks. IBM MQ for UNIX, Linux and Windows systems adopt the ITU X.509 guidance so that security will not be adversely reduced.

A potential scenario exists where the CA that issues a CRL might set an unknown critical extension to indicate that even though all other validation checks are successful, a certificate which is identified must not be considered revoked and thus not rejected by the application. In this scenario, following X.509, IBM MQ for UNIX, Linux and Windows systems will function in a fail-secure mode of operation. That is, they might reject certificates that the CA did not intend to be rejected and therefore might deny service to some valid users. A fail-insecure mode ignores a CRL because it has an unknown critical extension and therefore certificates that the CA intended to be revoked are still accepted. The administrator of the system should then query this behavior with the issuing CA.

- Excluded subtrees: a set of root names defining a set of subtrees within which no subject name in subsequent certificates in the certification path can fall. The initial value is "empty".
- Explicit policy: an integer which indicates if an explicit policy identifier is required. The integer indicates the first certificate in the path where this requirement is imposed. When set, this variable can be decreased, but cannot be increased. (That is, if a certificate in the path requires explicit policy identifiers, a later certificate cannot remove this requirement.) The initial value is *n*+1.
- Policy mapping: an integer which indicates if policy mapping is permitted. The integer indicates the last certificate on which policy mapping may be applied. When set, this variable can be decreased, but cannot be increased. (That is, if a certificate in the path specifies policy mapping is not permitted, it cannot be overridden by a later certificate.) The initial value is *n*+1.

The validation of a chain is performed in the following manner (but not necessarily the following order):

1. The information in the following paragraph is consistent with the basic path validation policy described in “Basic path validation policy” on page 16:

   Ensure that the name of the certificate's issuer is equal to the subject name in the previous certificate, and that there is not an empty issuer name in this certificate or the previous certificate subject name. If no previous certificate exists in the path and this is the first certificate in the chain, ensure that the issuer and subject name are identical and that the certificate is trusted.

   If the certificate does not have a subject name, the subjectAltName extension must be present and critical.
2. The information in the following paragraph is consistent with the basic path validation policy described in “Basic path validation policy” on page 16:

   Ensure that the signature algorithm used to actually sign the certificate matches the signature algorithm indicated within the certificate, by ensuring that the issuer signature algorithm identifier in the certificate matches the algorithm identifier in the signature data.

   If both the certificate's issuersUniqueID and the issuer's subjectUniqueID are present, ensure they match.
3. The following information is consistent with the basic path validation policy described in “Basic path validation policy” on page 16:

   Ensure that the certificate was signed by the issuer, using the subject public key from the previous certificate in the path to verify the signature on the certificate. If no previous certificate exists and this is the first certificate, use the subject public key of the certificate to verify the signature on it.
4. The following information is consistent with the basic path validation policy described in “Basic path validation policy” on page 16:

   Ensure that the certificate is a known X509 version, unique IDs are not present for version 1 certificates and extensions are not present for version 1 and version 2 certificates.
5. The following information is consistent with the basic path validation policy described in “Basic path validation policy” on page 16:

   Ensure that the certificate has not expired, or not been activated yet, and that its validity period is good [16]
6. The following information is consistent with the basic path validation policy described in “Basic path validation policy” on page 16:

   Ensure that there are no unknown critical extensions, nor any duplicate extensions.
7. The following information is consistent with the basic path validation policy described in “Basic path validation policy” on page 16:

   Ensure that the certificate has not been revoked. Here, the following operations apply:

[16] There are no checks to ensure the subject's validity is within bounds of the issuer's validity. This is not required, and certificates from some CAs have been shown to not pass such a check.

a. If the OCSP connection is enabled and a Responder Address is configured or the Certificate has a valid AuthorityInfoAccess extension specifying an HTTP format GENERALNAME_uniformResourceID check revocation status with OCSP.

   i) IBM MQ for UNIX and Windows systems allows the OCSP Request to be optionally signed for preconfigured responders but this has otherwise no impact on OCSP Response processing.

b. If revocation status from 7a is undetermined the CRLDistributionPoints extension is checked for a list of X.500 distinguished name GENERALNAME_directoryname and URI GENERALNAME_uniformResourceID. If the extension is not present, the certificate's issuer's name is used. A CRL database (LDAP) is then queried for CRLs. If the certificate is not the last certificate, or if the last certificate has the basic constraint extension with the "isCA" flag turned on, the database is queried for ARL's and CRL's instead. If CRL checking is enabled, and no CRL database can be queried, the certificate is treated as revoked. Currently, the X500 directory name form and the LDAP/HTTP/FILE URI forms are the only supported name forms used to look up CRLs and ARLs15.

   Note: RelativeDistinguishedNames are not supported.

8. The following information is consistent with the basic path validation policy described in “Basic path validation policy” on page 16:

   If the subjectAltName extension is marked critical, ensure that the name forms are recognized. The following general name forms are currently recognized:

   - rfc822
   - DNS
   - directory
   - URI
   - IPAddress(v4/v6)

9. Ensure that the subject name and subjectAltName extension (critical or noncritical) is consistent with the constrained and excluded subtrees state variables.
10. If the EmailAddress OID is present in the subject name field as an IA5 string, and there is no subjectAltName extension, the EmailAddress must be consistent with the constrained and excluded subtrees state variable.
11. Ensure that policy information is consistent with the initial policy set :

    a. If the explicit policy state variable is less than or equal to the current certificate's numeric sequence value, a policy identifier in the certificate shall be in the initial policy set.

    b. If the policy mapping variable is less than or equal to the current certificate's numeric sequence value, the policy identifier cannot be mapped.

12. Ensure that policy information is consistent with the acceptable policy set:

    a. If the certificate policies extension is marked critical [17] , the intersection of the policies extension and the acceptable policy set is non-null.

    b. The acceptable policy set is assigned the resulting intersection as its new value.

13. Ensure that the intersection of the acceptable policy set and the initial policy set is non-null. If the special Policy of anyPolicy is present then allow it only if it has not been inhibited by the inhibitAnyPolicy extension at this chain position.
14. If an inhibitAnyPolicy extension is present ensure that it is marked Critical and, if so, set the inhibitAnyPolicy state and chain position to the value of the integer value of the extension provided it is not greater than the current value. This is the number of certificates to allow with an anyPolicy Policy before disallowing the anyPolicy Policy.
15. The following steps are performed for all certificates except the last one:

[17] This is maintained as a legacy requirement from RFC2459 (6.1 (e)(1))

a. If the issuerAltName extension is marked critical, ensure that the name forms are recognized. The following general name forms are currently recognized:

- rfc822
- DNS
- directory
- URI
- IPAddress(v4/v6)

b. i) If the BasicConstraints extension is not present, the certificate is only valid as an EE certificate.

ii) If the BasicConstraints extension is present, ensure that the "isCA" flag is true. Note that "isCA" is always checked to ensure it is true to be as part of the chain building itself, however this specific test is still made. If the pathLength field is present, ensure the number of certificates until the last certificate is not greater than the pathLength field.

c. If the KeyUsage extension is critical, ensure that the keyCertSign flag is on, and ensure that if the BasicConstraints extension is present, that the "isCA" flag is true [18] .

d. If a policy constraints extension is included in the certificate, modify the explicit policy and policy mapping state variables as follows:

- i. If requireExplicitPolicy is present and has value *r*, the explicit policy state variable is set to the minimum of its current value and the sum of *r* and *i* (the current certificate in the sequence).
- ii. If inhibitPolicyMapping is present and has value *q*, the policy mapping state variable is set to the minimum of its current value and the sum of *q* and *i* (the current certificate in the sequence).

e. If the policyMappings extension is present (see 12(b)), ensure that it is not critical, and if policy mapping is allowed, these mappings are used to map between this certificate's policies and its signee's policies.

f. If the nameConstraints extension is present , ensure that it is critical, and that the permitted and excluded subtrees adhere to the following rules before updating the chain's subtree's state in accordance with the algorithm described in RFC 5280 section 6.1.4 part (g):

i) The minimum field is set to zero.

ii) The maximum field is not present.

iii) The base field name forms are recognized. The following general name forms are currently recognized:

- rfc822
- DNS
- directory
- URI
- IPAddress(v4/v6)

16. V 10.0.0 The ExtendedKeyUsage extension is checked by IBM MQ unless **RFC5280EKUChecks=FALSE** is set in the SSL stanza of the configuration files. For more information see Certificate Extended Key Usage (EKU).

    LTS The ExtendedKeyUsage extension is not checked by IBM MQ.

17. The following information is consistent with the basic path validation policy described in “Basic path validation policy” on page 16:

    The AuthorityKeyID extension is not used for path validation, but is used when building the certificate chain.

18. The following information is consistent with the basic path validation policy described in “Basic path validation policy” on page 16:

[18] This check is in fact redundant because of step (b), but the check is still made.

The SubjectKeyID extension is not used for path validation, but is used when building the certificate chain.

19. The following information is consistent with the basic path validation policy described in “Basic path validation policy” on page 16:

    The PrivateKeyUsagePeriod extension is ignored by the validation engine, because it cannot determine when the CA actually signed the certificate. The extension is always non-critical and therefore can be safely ignored.

Managed File Transfer security reference

Reference information to help you configure security for Managed File Transfer.

File system permissions for MFT in IBM MQ

When you install and configure the Managed File Transfer component of IBM MQ, the `configuration`, `installations`, and `logs` directories are created with the following permissions.

AIX and Linux

Linux AIX

Table 1. Summary of permissions for directories on AIX and Linux

Directory	Permissions
`/var/mqm/mqft/config`	• Writable by the mqm group • World readable Users in the mqm group have write access to these directories and files The directory has the SetGroupId bit 's' set, meaning that all files created in it have the group ownership of mqm.
`/var/mqm/mqft/installations`	• Writable by the mqm group • World readable The directory has the SetGroupId bit 's' set, meaning that all files created in it have the group ownership of mqm.
`/var/mqm/mqft/logs`	World readable and writable The directory has the SetGroupId bit 's' set, meaning that all files created in it have the group ownership of mqm.

Windows

Windows

Table 2. Summary of permissions for directories on Windows	
Directory	**Permissions**
MQ_DATA_PATH\mqft\config	The following users have full read and write access: • Administrators • System account • mqm group Other users have read access
MQ_DATA_PATH\mqft\installations	The following users have full read and write access: • Administrators • System account • mqm group Other users have read access
MQ_DATA_PATH\mqft\logs	The following users have full read and write access: • Administrators • System account • mqm group Other users have read and write access

z/OS

z/OS

Table 3. Summary of permissions for directories on z/OS	
Directory	**Permissions**
DATA_PATH/mqft/config	• Writable by the mqm group, or the group name identified in the environment variable BFG_GROUP_NAME • World readable Users in the mqm group, or the value in the environment variable BFG_GROUP_NAME, have write access to these directories and files
DATA_PATH/mqft/installations	• Writable by the mqm group, or the group name identified in the environment variable BFG_GROUP_NAME • World readable
DATA_PATH/mqft/logs	World readable and writable

Note: *DATA_PATH* is derived from the environment variable BFG_DATA.

Managing authorities for MFT-specific resources

For any file transfer request, the Managed File Transfer Agent processes require some level of access to their local file systems. In addition, both the user identifier associated with the agent process, and the

user identifiers associated with users performing file transfer operations must have the authority to use certain IBM MQ objects.

Commands are issued by users, who might be in an operational role where they typically start a file transfer. Alternatively, they might be in an administrative role where they can additionally control when agents are created, started, deleted, or cleaned (that is, when messages from all agent system queues are removed). Messages containing command requests are placed on an agent's SYSTEM.FTE.COMMAND queue when a user issues a command. The agent process retrieves messages containing command requests from the SYSTEM.FTE.COMMAND queue. The agent process also uses four other system queues, which are as follows:

- SYSTEM.FTE.DATA.*agent_name*
- SYSTEM.FTE.EVENT.*agent_name*
- SYSTEM.FTE.REPLY.*agent_name*
- SYSTEM.FTE.STATE.*agent_name*

Because users issuing commands use the queues listed previously in different ways to the agent process, assign different IBM MQ authorities to the user identifiers or user groups associated with each. See “Restricting group authorities for MFT-specific resources” on page 26 for more information.

The agent has additional queues that can be used to grant users the authority to perform certain actions. See “Restricting user authorities on MFT agent actions” on page 31 for information about how to use the authority queues. The agent does not put or get messages on these queues. However, you must ensure that the queues are assigned the correct IBM MQ authorities both for the user identifier used to run the agent process as well as the user identifiers associated with users who are being authorized to perform certain actions. The authority queues are as follows:

- SYSTEM.FTE.AUTHADM1.*agent_name*
- SYSTEM.FTE.AUTHAGT1.*agent_name*
- SYSTEM.FTE.AUTHMON1.*agent_name*
- SYSTEM.FTE.AUTHOPS1.*agent_name*
- SYSTEM.FTE.AUTHSCH1.*agent_name*
- SYSTEM.FTE.AUTHTRN1.*agent_name*

If you are migrating from a version of Managed File Transfer earlier than version 7.0.2 to IBM WebSphere® MQ 7.5, or later, and are keeping existing agent configurations, you will need to create the authority queues manually. Use the following MQSC command to create the queues:

```
DEFINE QLOCAL(authority_queue_name) DEFPRTY(0) DEFSOPT(SHARED) GET(ENABLED) MAXDEPTH(0) +
 MAXMSGL(0) MSGDLVSQ(PRIORITY) PUT(ENABLED) RETINTVL(999999999) SHARE  NOTRIGGER +
 USAGE(NORMAL) REPLACE
```

The agent process also publishes messages to the SYSTEM.FTE topic on the coordination queue manager using the SYSTEM.FTE queue. Depending on whether the agent process is in the role of the source agent or destination agent, the agent process might require authority to read, write, update, and delete files.

You can create and modify authority records for IBM MQ objects using the IBM MQ Explorer. Right-click the object and then click **Object Authorities** > **Manage Authority Records**. You can also create authority records using the **setmqaut** command, which is described at setmqaut (grant or revoke authority) command.

Related reference

“Restricting group authorities for MFT-specific resources” on page 26
Instead of granting authority to individual users for all of the various objects that might be involved, configure two security groups for the purposes of administering Managed File Transfer access control: FTEUSER and FTEAGENT. It is the responsibility of the IBM MQ administrator to create and populate these groups. The administrator can choose to extend or modify the proposed configuration described here.

“Restricting user authorities on MFT agent actions” on page 31
In addition to using groups to manage access to resources, you can enable an additional level of security to restrict the Managed File Transfer agent actions that a user can take. Grant authorities on an agent authority queue to a user to give the user permission to perform specific agent actions.

Authorities for the MFT logger

Restricting group authorities for MFT-specific resources

Instead of granting authority to individual users for all of the various objects that might be involved, configure two security groups for the purposes of administering Managed File Transfer access control: FTEUSER and FTEAGENT. It is the responsibility of the IBM MQ administrator to create and populate these groups. The administrator can choose to extend or modify the proposed configuration described here.

Authority to connect to queue managers

Commands that are run by operational users, administrative users, and the IBM MQ Explorer need to be able to connect to the command queue manager and coordination queue manager. The agent process and commands that are run to create, alter, or delete the agent need to be able to connect to the agent queue manager.

- Grant the FTEUSER group connect authority for the command queue manager and coordination queue manager. For example:

  ALW **For AIX, Linux, and Windows systems:**

  ```
  setmqaut -m command_queue_manager -t qmgr -g FTEUSER +connect
  setmqaut -m coordination_queue_manager -t qmgr -g FTEUSER +connect
  ```

  IBM i **For IBM i:**

  ```
  GRTMQMAUT OBJ('command_queue_manager') OBJTYPE(*MQM) USER(FTEUSER) AUT(*CONNECT)
  GRTMQMAUT OBJ('coordination_queue_manager') OBJTYPE(*MQM) USER(FTEUSER) AUT(*CONNECT)
  ```

  z/OS **For z/OS:**

  ```
  RDEFINE MQCONN command_queue_manager.BATCH UACC(NONE)
  PERMIT command_queue_manager.BATCH CLASS(MQCONN) ID(FTEUSER) ACCESS(READ)
  RDEFINE MQCONN coordination_queue_manager.BATCH UACC(NONE)
  PERMIT coordination_queue_manager.BATCH CLASS(MQCONN) ID(FTEUSER) ACCESS(READ)
  ```

- Grant the FTEAGENT group connect and inquire authority to the agent queue manager. For example:

  ALW **For AIX, Linux, and Windows systems:**

  ```
  setmqaut -m agent_queue_manager -t qmgr -g FTEAGENT +connect +inq +setid
  ```

  IBM i **For IBM i:**

  ```
  GRTMQMAUT OBJ('agent_queue_manager') OBJTYPE(*MQM) USER(FTEAGENT) AUT(*CONNECT)
  ```

  z/OS **For z/OS:**

  ```
  RDEFINE MQCONN agent_queue_manager.BATCH UACC(NONE)
  PERMIT agent_queue_manager.BATCH CLASS(MQCONN) ID(FTEAGENT) ACCESS(READ)
  ```

For information about which command directly connects to which queue manager, see Which MFT command connects to which queue manager

Authority to put a message on the COMMAND queue that belongs to the agent

The agent command queue must be available to any user who is authorized to request that the agent performs an action. To satisfy this requirement,

- Grant the FTEUSER group only put access to the SYSTEM.FTE.COMMAND.*agent_name* queue. For example:

  ALW **For AIX, Linux, and Windows systems:**

  ```
  setmqaut -m QM1 -n SYSTEM.FTE.COMMAND.agent_name -t queue -g FTEUSER +put
  ```

  IBM i **For IBM i:**

  ```
  GRTMQMAUT OBJ('SYSTEM.FTE.COMMAND.agent_name') OBJTYPE(*Q) USER(FTEUSER) AUT(*PUT)
  MQMNAME('QM1')
  ```

  z/OS **For z/OS:**

  ```
  RDEFINE MQQUEUE QM1.SYSTEM.FTE.COMMAND.agent_name UACC(NONE)
  PERMIT QM1.SYSTEM.FTE.COMMAND.agent_name CLASS(MQQUEUE) ID(FTEUSER) ACCESS(UPDATE)
  ```

- Grant the FTEAGENT group put, get, and setid access to the SYSTEM.FTE.COMMAND.*agent_name* queue. For example:

  ALW **For AIX, Linux, and Windows systems:**

  ```
  setmqaut -m QM1 -n SYSTEM.FTE.COMMAND.agent_name -t queue -g FTEAGENT +browse +put +get
  +setid
  ```

  IBM i **For IBM i:**

  ```
  GRTMQMAUT OBJ('SYSTEM.FTE.COMMAND.agent_name') OBJTYPE(*Q) USER(FTEAGENT) AUT(*PUT)
  MQMNAME('QM1')
  GRTMQMAUT OBJ('SYSTEM.FTE.COMMAND.agent_name') OBJTYPE(*Q) USER(FTEAGENT) AUT(*GET)
  MQMNAME('QM1')
  GRTMQMAUT OBJ('SYSTEM.FTE.COMMAND.agent_name') OBJTYPE(*Q) USER(FTEAGENT) AUT(*SETID)
  MQMNAME('QM1')
  ```

  z/OS **For z/OS:**

  ```
  RDEFINE MQQUEUE QM1.SYSTEM.FTE.COMMAND.agent_name UACC(NONE)
  PERMIT QM1.SYSTEM.FTE.COMMAND.agent_name CLASS(MQQUEUE) ID(FTEAGENT) ACCESS(UPDATE)
  RDEFINE MQADMIN QM1.CONTEXT.SYSTEM.FTE.COMMAND.agent_name UACC(NONE)
  PERMIT QM1.CONTEXT.SYSTEM.FTE.COMMAND.agent_name CLASS(MQADMIN) ID(FTEAGENT)
  ACCESS(UPDATE)
  ```

Agents need access to put messages to other agents' command queues. If there are agents connected to remote queue managers, you might need to grant additional authorization to allow the channel to put messages to this queue.

Authority to put messages on the DATA, STATE, EVENT, and REPLY queues that belong to the agent

Only Managed File Transfer agents need to be able to use these system queues, therefore grant the group FTEAGENT put, get and inquire access. The names of these system queues are as follows:

- DATA - SYSTEM.FTE.DATA.*agent_name*
- STATE - SYSTEM.FTE.STATE.*agent_name*
- EVENT - SYSTEM.FTE.EVENT.*agent_name*
- REPLY - SYSTEM.FTE.REPLY.*agent_name*

For example, for the SYSTEM.FTE.DATA.*agent_name* queue, use a command like the following:

For AIX, Linux, and Windows systems:

```
setmqaut -m QM1 -n SYSTEM.FTE.DATA.agent_name -t queue -g FTEAGENT +put +get +inq
```

For IBM i:

```
GRTMQMAUT OBJ('SYSTEM.FTE.DATA.agent_name') OBJTYPE(*Q) USER(FTEAGENT) AUT(*PUT)
MQMNAME('QM1')
GRTMQMAUT OBJ('SYSTEM.FTE.DATA.agent_name') OBJTYPE(*Q) USER(FTEAGENT) AUT(*GET)
MQMNAME('QM1')
```

For z/OS:

```
RDEFINE MQQUEUE QM1.SYSTEM.FTE.DATA.agent_name UACC(NONE)
PERMIT QM1.SYSTEM.FTE.DATA.agent_name CLASS(MQQUEUE) ID(FTEAGENT) ACCESS(UPDATE)
```

Agents need access to put messages to other agents' data and reply queues. If there are agents connected to remote queue managers, you might need to grant additional authorization to allow the channel to put messages to these queues.

Authority that the agent process runs under

The authority that the agent process runs under affects the files the agent can read and write from the file system, and the queues and topics the agent can access. How the authority is configured is system-dependent. Add the user ID that the agent process runs under to the FTEAGENT group. For more information about adding a user ID to a group, see Setting up security and navigate to the information for your operating system.

Authority that the commands and IBM MQ Explorer run under

Administrative commands, for example the **fteStartAgent** command, and the Managed File Transfer plug-in for the IBM MQ Explorer need to be able to put messages to the SYSTEM.FTE.COMMAND.*agent_name* queue and retrieve published information from that queue. Add the user IDs that are authorized to run the commands or the IBM MQ Explorer to the FTEUSER group. This originator user ID is recorded in the transfer log. For more information about adding a user ID to a group, see Setting up security and navigate to the information for your operating system.

Authority to put messages on the SYSTEM.FTE queue and SYSTEM.FTE topic

Only the agent process needs to be able to place messages on the SYSTEM.FTE queue and SYSTEM.FTE topic. Grant put, get and inquire authority to the FTEAGENT group on the SYSTEM.FTE queue, and grant publish and subscribe authority to the FTEAGENT group on the SYSTEM.FTE topic. For example:

For AIX, Linux, and Windows systems:

```
setmqaut -m QM1 -n SYSTEM.FTE -t queue -g FTEAGENT +put +inq
setmqaut -m QM1 -n SYSTEM.FTE -t topic -g FTEAGENT +pub +sub +resume
```

For IBM i:

```
GRTMQMAUT OBJ('SYSTEM.FTE') OBJTYPE(*Q) USER(FTEAGENT) AUT(*PUT) MQMNAME('QM1')
GRTMQMAUT OBJ('SYSTEM.FTE') OBJTYPE(*TOPIC) USER(FTEAGENT) AUT(*PUB) MQMNAME('QM1')
GRTMQMAUT OBJ('SYSTEM.FTE') OBJTYPE(*TOPIC) USER(FTEAGENT) AUT(*SUB) MQMNAME('QM1')
```

For z/OS:

```
RDEFINE MQQUEUE QM1.SYSTEM.FTE UACC(NONE)
PERMIT QM1.SYSTEM.FTE CLASS(MQQUEUE) ID(FTEAGENT) ACCESS(UPDATE)
RDEFINE MXTOPIC QM1.PUBLISH.SYSTEM.FTE UACC(NONE)
PERMIT QM1.PUBLISH.SYSTEM.FTE CLASS(MXTOPIC) ID(FTEAGENT) ACCESS(UPDATE)
```

If there are agents connected to remote queue managers, additional authorization might also need to be granted to allow the channel to put messages to the SYSTEM.FTE queue.

For a message to get published to the SYSTEM.FTE topic, the authority records of the SYSTEM.FTE topic must allow publication by the user ID contained in the message descriptor structure (MQMD) of the message. This is described in “Authority to publish MFT Agents log and status messages” on page 38.

z/OS To allow a user to publish to the SYSTEM.FTE topic on z/OS, you must grant the channel initiator user ID access to publish to the SYSTEM.FTE topic. If the RESLEVEL security profile causes two user IDs to be checked for the channel initiator connection, you also need to grant access to the user ID contained in the message descriptor structure (MQMD) of the message. For more information, see The RESLEVEL security profile

Authority to receive publications on the SYSTEM.FTE topic

Transfer log messages, progress messages, and status messages are intended for general use, so grant the FTEUSER group authority to subscribe to the SYSTEM.FTE topic. For example:

ALW **For AIX, Linux, and Windows systems:**

```
setmqaut -m QM1 -n SYSTEM.FTE -t topic -g FTEUSER +sub
```

IBM i **For IBM i:**

```
GRTMQMAUT OBJ('SYSTEM.FTE') OBJTYPE(*TOPIC) USER(FTEUSER) AUT(*SUB) MQMNAME('QM1')
```

z/OS **For z/OS:**

```
RDEFINE MXTOPIC QM1.SUBSCRIBE.SYSTEM.FTE UACC(NONE)
PERMIT QM1.SUBSCRIBE.SYSTEM.FTE CLASS(MXTOPIC) ID(FTEUSER) ACCESS(ALTER)
```

Authority to connect to remote queue managers using transmission queues

In a topology of multiple queue managers, the agent requires put authority on the transmission queues used to connect to the remote queue managers.

Authority to create a temporary reply queue for file transfers

File transfer requests wait for the transfer to complete and rely on a temporary reply queue being created and populated. Grant the FTEUSER group DISPLAY, PUT, GET, and BROWSE authorities on the temporary model queue definition. For example:

ALW **For AIX, Linux, and Windows systems:**

```
setmqaut -m QM1 -n SYSTEM.DEFAULT.MODEL.QUEUE -t queue -g FTEUSER +dsp +put +get +browse
```

IBM i **For IBM i:**

```
GRTMQMAUT OBJ('SYSTEM.DEFAULT.MODEL.QUEUE') OBJTYPE(*Q) USER(FTEUSER) AUT(*ADMDSP)
MQMNAME('QM1')
GRTMQMAUT OBJ('SYSTEM.DEFAULT.MODEL.QUEUE') OBJTYPE(*Q) USER(FTEUSER) AUT(*PUT)
MQMNAME('QM1')
GRTMQMAUT OBJ('SYSTEM.DEFAULT.MODEL.QUEUE') OBJTYPE(*Q) USER(FTEUSER) AUT(*GET)
MQMNAME('QM1')
GRTMQMAUT OBJ('SYSTEM.DEFAULT.MODEL.QUEUE') OBJTYPE(*Q) USER(FTEUSER) AUT(*BROWSE)
MQMNAME('QM1')
```

z/OS **For z/OS:**

```
RDEFINE MQQUEUE QM1.SYSTEM.DEFAULT.MODEL.QUEUE UACC(NONE)
PERMIT QM1.SYSTEM.DEFAULT.MODEL.QUEUE CLASS(MQQUEUE) ID(FTEUSER) ACCESS(UPDATE)
```

By default, this queue is SYSTEM.DEFAULT.MODEL.QUEUE, but you can configure the name by setting values for the properties 'modelQueueName' and 'dynamicQueuePrefix' in the `command.properties` file.

z/OS On z/OS, you must also grant authority to access the temporary queues to FTEUSER. For example:

```
RDEFINE MQQUEUE QM1.WMQFTE.** UACC(NONE)
PERMIT QM1.WMQFTE.** CLASS(MQQUEUE) ID(FTEUSER) ACCESS(UPDATE)
```

By default the name of each temporary queue on z/OS starts with WMQFTE.

The following table summarizes the access control configuration for FTEUSER and FTEAGENT in the security scheme described:

Table 4. Summary of access control configuration for FTEUSER and FTEAGENT

Object	**Object type**	**FTEUSER**	**FTEAGENT**
Agent queue manager	Queue manager		CONNECT, INQ, and SETID. ALTUSER is also required to enable "Restricting user authorities on MFT agent actions" on page 31.
Coordination queue manager	Queue manager		
Command queue manager	Queue manager	CONNECT	CONNECT
SYSTEM.FTE	Local queue		GET and PUT
SYSTEM.FTE.COMMAND.*agent_name*	Local queue	PUT	BROWSE, GET, PUT, and SETID
SYSTEM.FTE.DATA.*agent_name*	Local queue		GET and PUT
SYSTEM.FTE.EVENT.*agent_name*	Local queue		BROWSE, GET and PUT
SYSTEM.FTE.REPLY.*agent_name*	Local queue		GET and PUT
SYSTEM.FTE.STATE.*agent_name*	Local queue		BROWSE, GET, INQ, and PUT
SYSTEM.FTE	Local topic	SUBSCRIBE	PUBLISH and SUBSCRIBE
SYSTEM.DEFAULT.MODEL.QUEUE (or the model queue defined in Managed File Transfer that is used to create a temporary reply queue.)	Model queue	BROWSE, DISPLAY, GET, and PUT	BROWSE, DISPLAY, GET, and PUT
Transmission queues to communicate with remote queue managers	Local queue		PUT

Authority to manage transfers through IBM MQ Explorer

In addition to granting MFT authorities to users in situations that are already mentioned on this page, further authorities need to be granted to the MFT agent user who administers and performs all MFT operations through IBM MQ Explorer. To issue commands such as create, cancel, schedule file transfer, create, delete resource monitors, and create transfer templates, the IBM MQ Explorer user must have authority as follows:

- Coordination queue manager: connect, inquire, display
- Command queue manager: connect, inquire, display
- SYSTEM.FTE topic: publish, subscribe
- SYSTEM.MQEXPLORER.REPLY.MODEL: display, inquire, get, browse, put
- SYSTEM.ADMIN.COMMAND.QUEUE: inquire, put, display
- SYSTEM.DEFAULT.MODEL.QUEUE: get, put, inquire, display, browse

See also Which MFT command connects to which queue manager.

Related reference

"Restricting user authorities on MFT agent actions" on page 31
In addition to using groups to manage access to resources, you can enable an additional level of security to restrict the Managed File Transfer agent actions that a user can take. Grant authorities on an agent authority queue to a user to give the user permission to perform specific agent actions.

Authorities for the MFT logger

Restricting user authorities on MFT agent actions

In addition to using groups to manage access to resources, you can enable an additional level of security to restrict the Managed File Transfer agent actions that a user can take. Grant authorities on an agent authority queue to a user to give the user permission to perform specific agent actions.

Enabling user authority management

To turn on user authority checking on agent actions, complete the following steps:

1. In the `agent.properties` file, set the `authorityChecking` value to `true`. For more information, see The MFT `agent.properties` file.
2. Ensure that the user who runs the agent has the IBM MQ alternate user (ALTUSER) authority to the agent queue manager.

   z/OS On the z/OS platform, the user that runs the agent must have ALTUSER authority to the user IDs that can request permission to perform an agent action.

Both agents involved in a transfer must have the same level of security enabled, that is, `authorityChecking` must be set to the same value in the property files of both agents. Transfers between agents that have different values for the `authorityChecking` property will fail.

Agent authority queues

The agent has authority queues that are used to manage which users have the authority to perform certain agent actions. The agent does not put or get messages to these queues. The agent authority queues are as follows:

- SYSTEM.FTE.AUTHADM1.*agent_name*
- SYSTEM.FTE.AUTHAGT1.*agent_name*
- SYSTEM.FTE.AUTHMON1.*agent_name*
- SYSTEM.FTE.AUTHOPS1.*agent_name*
- SYSTEM.FTE.AUTHSCH1.*agent_name*
- SYSTEM.FTE.AUTHTRN1.*agent_name*

When user authority management is enabled by setting the agent property **`authorityChecking=true`**, the authorities that a user has on the agent authority queues specify the actions that the user is authorized to take.

Important: The `inquire` permission is a required permission on all of the agent authority queues.

The following table summarizes the IBM MQ access authorities that users or groups require, in addition to `inquire` permission, on an agent authority queue to perform specific actions.

Table 5. The level of IBM MQ access authority that a user or group requires on an agent authority queue to perform specific actions.

User action	Managed File Transfer access authority	Authority queues	IBM MQ access authority (Multiplatforms)	RACF® access level (z/OS only)
Shut down the agent, using the `-m` option on **`fteStopAgent`** command.	Administration	SYSTEM.FTE.AUTHADM1.*agent_name*	BROWSE	READ
Start a transfer of files from this agent	Transfer source	SYSTEM.FTE.AUTHTRN1.*source_agent_name*	BROWSE	READ
Run a managed call on this agent				
Cancel a transfer of files from this agent started by the same user				
Start a transfer of files to this agent	Transfer destination	SYSTEM.FTE.AUTHTRN1.*destination_agent_name*	PUT	UPDATE
Cancel a transfer of files to this agent started by the same user				
Create a resource monitor	Monitor	SYSTEM.FTE.AUTHMON1.*monitor_agent_name*	BROWSE	READ
Delete a resource monitor created by the same user				
Delete a resource monitor created by any user	Monitor operations	SYSTEM.FTE.AUTHOPS1.*agent_name*	SET	ALTER
Create a scheduled transfer	Schedule	SYSTEM.FTE.AUTHSCH1.*source_agent_name*	BROWSE	READ
Delete a scheduled transfer created by the same user				
Delete a scheduled transfer created by any user or group	Schedule operations	SYSTEM.FTE.AUTHOPS1.*agent_name*	PUT	UPDATE
Cancel a transfer created either by the same user or group that started the transfer, or by another user or group	Transfer operations	SYSTEM.FTE.AUTHOPS1.*source_agent_name* SYSTEM.FTE.AUTHOPS1.*destination_agent_name*	BROWSE	READ

Note: To give a user or group permission to set up a resource monitor or scheduled transfer that starts a transfer the user needs both the `Monitor` or `Schedule` authority and `Transfer` source and `Transfer` destination authorities.

When an agent receives a request to cancel a file transfer, the agent first checks whether the user requesting the cancellation is the same user who started the transfer. If the user canceling the transfer is not the same as the user that requested it, the agent then checks if the canceling user has `browse` permission on SYSTEM.FTE.AUTHOPS1.*agent_name* queues. Checking authority in this order avoids

unexpected errors in agent and queue manager error logs when the user who requested the file transfer and the user who requested the cancellation are the same.

A user can start one agent and want it to interact with another agent. How the two agents can interact depends on the level of access authority that the user has on the other agent authority queue.

Table 6. The level of IBM MQ access authority that the user that starts an agent requires on another agent authority queue so that files can be transferred between the agents.

Agent action	Managed File Transfer access authority	Authority queues	IBM MQ access authority (Multiplatforms)	RACF access level (z/OS only)
Receive a transfer from *source_agent*	Agent source	SYSTEM.FTE.AUTHAGT1.*source_agent_name*	BROWSE	READ
Send a transfer to *destination_agent*	Agent destination	SYSTEM.FTE.AUTHAGT1.*destination_agent_name*	PUT	UPDATE
Start a managed call from an agent	Agent source	SYSTEM.FTE.AUTHTRN1.*agent_name*	BROWSE	READ

Configuring user authority management

To authorize a user to be able to perform an action on an agent, grant the user the appropriate authority on the relevant authority queue. To grant authorities to a user, complete the following steps:

1. Create a user on the system where the agent queue manager is located that has the same name as the user you want to give authority to perform agent actions. This user does not have to be active.
2. Grant the user the appropriate authority on the relevant authority queue. If you are using AIX, Linux, or Windows, you can use the setmqaut command.
3. Refresh the security configuration of the queue manager. You can use the REFRESH SECURITY MQSC command.

Example

z/OS The **setmqaut** command is not used on z/OS systems. For z/OS, instead use RACF. See Setting up security on z/OS for more information.

IBM i The **setmqaut** command is not used on IBM i systems. For IBM i, see Access authorities for IBM MQ objects, which describes how authorization for IBM MQ objects is done. There are three relevant CL commands available on IBM i: **Grant MQ Object Authority (GRTMQMAUT)**, **Revoke MQ Object Authority (RVKMQMAUT)**, and **Refresh MQ Authority (RFRMQMAUT)**.

A user, who is a member of the group `requestor_group`, wants to set up a resource monitor on AGENT1 that transfers a file from AGENT1, which is running under the user `user1`, who is a member of the group `user1_group`, to AGENT2, which is running under the user `user2`, who is a member of the group `user2_group`. AGENT1 connects to QM1; AGENT2 connects to QM2. Both agents have authority checking enabled. To make this possible take the following steps:

1. `requestor` must have **Monitor** authority on AGENT1. Set this authority by running the following command on the system where QM1 is running:

```
setmqaut -m QM1 -t queue -n SYSTEM.FTE.AUTHMON1.AGENT1 -g requestor_group +browse
```

2. `requestor` must have **Transfer source** authority on AGENT1. Set this authority by running the following command on the system where QM1 is running:

```
setmqaut -m QM1 -t queue -n SYSTEM.FTE.AUTHTRN1.AGENT1 -g requestor_group +browse
```

3. `requestor` must have **Transfer destination** authority on AGENT2. Set this authority by running the following command On the system where QM2 is running:

```
setmqaut -m QM2 -t queue -n SYSTEM.FTE.AUTHTRN1.AGENT2 -g requestor_group +put
```

4. `user2` must have **Agent source** authority on AGENT1. Set this authority by running the following command on the system where QM1 is running:

```
setmqaut -m QM1 -t queue -n SYSTEM.FTE.AUTHAGT1.AGENT1 -g user2_group +browse
```

5. `user1` must have **Agent destination** authority on AGENT2. Set this authority by running the following command on the system where QM2 is running:

```
setmqaut -m QM2 -t queue -n SYSTEM.FTE.AUTHAGT1.AGENT2 -g user1_group +put
```

Logging

If user authority checking is enabled, failed authority checks cause a not authorized log message to be published to the coordination queue manager. See MFT message formats for security for more information.

Messages about user authority can be written to the agent event log. You can configure the amount of information written to the agent event log by setting the `logAuthorityChecks` property in the agent property file. By default the level of authority check logging is `None`. You can also set the value of `logAuthorityChecks` to `Failures`, which specifies that only failed authorization checks are reported, or `All` which specifies that failed and successful authorization checks are reported.

For more information, see The MFT `agent.properties` file.

Related reference

“Restricting group authorities for MFT-specific resources” on page 26
Instead of granting authority to individual users for all of the various objects that might be involved, configure two security groups for the purposes of administering Managed File Transfer access control: FTEUSER and FTEAGENT. It is the responsibility of the IBM MQ administrator to create and populate these groups. The administrator can choose to extend or modify the proposed configuration described here.

Authorities for the MFT logger
fteStopAgent

MFT permissions to access sensitive configuration information

Any file used to store sensitive configuration information, meaning any file referenced from the IBM MQ configuration tree, must not have system-wide read, write, or (where applicable), delete permissions. These restrictions also apply to truststore and keystore files.

If a Managed File Transfer process detects a condition that a configuration file contains sensitive information, is a keystore or truststore file, and has system-wide read, write, or delete permissions, the process takes one of the following actions:

- Fails to start, if the condition is detected at startup time.
- Generates a warning message and ignores the contents of the configuration file if the condition was detected at runtime. This is relevant to the protocol bridge and the Connect:Direct® bridge, which reload a configuration if it changes while the process is running.

On systems with a UNIX type file system

The criteria for determining that a file has unacceptable system-wide permissions are:

- The others class has been granted read permission on the file
- The others class has been granted write permission on the file
- The others class has been granted write permission on the directory containing the file

On Windows systems

The criteria for determining that a file has unacceptable system-wide permissions are:

- Any of the Everyone, Guests, or Users groups have any of the following permissions:
  - Read data permission on the file
  - Append data permission on the file
  - Write data permission on the file
- Any of the Everyone, Guests, or Users groups has Create files permission on the folder containing the file and they also have any of the following permissions:
  - Delete subfolders and files permission on the folder containing the file
  - Delete permission on the file

Granting user authorities for MFT-specific resources

What authorities you need to provide to individual users, for all of the various objects, when you are connecting to the queue manager in the client transport.

Note that in the following examples, Bob is the name of the user attempting to connect, and queue manager MQ2 acts as the coordination, command, and agent queue managers

Authority to connect to queue managers

Commands that are run by operational users, administrative users, and the IBM MQ Explorer need to be able to connect to the command queue manager and coordination queue manager. The agent process and commands that are run to create, alter, or delete the agent need to be able to connect to the agent queue manager. See Which MFT commands and processes connect to which queue manager for more details.

The user running the commands, and starting the agent process, needs to have authority to connect to the relevant queue manager.

For AIX, Linux, and Windows systems:

```
setmqaut -m MQ2 -t qmgr -p Bob +connect +inq +setid
```

For IBM i:

```
GRTMQMAUT OBJ('MQ2') OBJTYPE(*MQM) USER(Bob) AUT(*CONNECT)
GRTMQMAUT OBJ('MQ2') OBJTYPE(*MQM) USER(Bob) AUT(*INQ)
GRTMQMAUT OBJ('MQ2') OBJTYPE(*MQM) USER(Bob) AUT(*SETID)
```

For z/OS:

```
RDEFINE MQCONN MQ2.BATCH UACC(NONE)
PERMIT MQ2.BATCH CLASS(MQCONN) ID(Bob) ACCESS(READ)
```

Authorities on the command queue that belongs to the agent

The user that the agent process is running as, and the user running commands that interact with the agent, needs put, get, set identity, and browse authorities on the SYSTEM.FTE.COMMAND.`agent_name` queue on the agent queue manager.

Note: In the following examples, you must replace `agent_name` with the actual name of the agent that you are using for the system queues.

ALW For AIX, Linux, and Windows systems:

```
setmqaut -m MQ2 -t queue -n SYSTEM.FTE.COMMAND.agent_name -m MQ2 -p Bob
+put +get +setid +browse
```

IBM i For IBM i:

```
GRTMQMAUT OBJ('SYSTEM.FTE.COMMAND.agent_name') OBJTYPE(*Q) USER(Bob)
AUT(*PUT) MQMNAME('MQ2')
GRTMQMAUT OBJ('SYSTEM.FTE.COMMAND.agent_name') OBJTYPE(*Q) USER(Bob)
AUT(*GET) MQMNAME('MQ2')
GRTMQMAUT OBJ('SYSTEM.FTE.COMMAND.agent_name') OBJTYPE(*Q) USER(Bob)
AUT(*SETID) MQMNAME('MQ2')
```

z/OS For z/OS:

```
RDEFINE MQQUEUE MQ2.SYSTEM.FTE.COMMAND.agent_name UACC(NONE)
PERMIT MQ2.SYSTEM.FTE.COMMAND.agent_name CLASS(MQQUEUE) ID(Bob) ACCESS(UPDATE)
RDEFINE MQADMIN MQ2.CONTEXT.SYSTEM.FTE.COMMAND.agent_name UACC(NONE)
PERMIT MQ2.CONTEXT.SYSTEM.FTE.COMMAND.agent_name CLASS(MQADMIN) ID(Bob) ACCESS(UPDATE)
```

Authority to put messages on the DATA, STATE, EVENT, and REPLY queues that belong to the agent

Put, get, and inquire access is required on the state queue, and put and get access to the other queues, on the agent queue manager.

Note: In the following examples, you must replace agent_name with the actual name of the agent that you are using for the system queues.

ALW For AIX, Linux, and Windows systems:

```
setmqaut -m MQ2 -t queue -n SYSTEM.FTE.REPLY.agent_name -m MQ2 -p Bob +put +get
setmqaut -m MQ2 -t queue -n SYSTEM.FTE.EVENT.agent_name -m MQ2 -p Bob +put +get +browse
setmqaut -m MQ2 -t queue -n SYSTEM.FTE.DATA.agent_name -m MQ2 -p Bob +put +get
setmqaut -m MQ2 -t queue -n SYSTEM.FTE.STATE.agent_name -m MQ2 -p Bob +put +get +inq
```

IBM i For IBM i:

```
AUT(*PUT) MQMNAME('MQ2')
GRTMQMAUT OBJ('SYSTEM.FTE.DATA.agent_name') OBJTYPE(*Q) USER(Bob) AUT(*GET) MQMNAME('MQ2')

GRTMQMAUT OBJ('SYSTEM.FTE.REPLY.agent_name') OBJTYPE(*Q) USER(Bob) AUT(*PUT) MQMNAME('MQ2')
GRTMQMAUT OBJ('SYSTEM.FTE.REPLY.agent_name') OBJTYPE(*Q) USER(Bob) AUT(*GET) MQMNAME('MQ2')

GRTMQMAUT OBJ('SYSTEM.FTE.EVENT.agent_name') OBJTYPE(*Q) USER(Bob) AUT(*PUT) MQMNAME('MQ2')
GRTMQMAUT OBJ('SYSTEM.FTE.EVENT.agent_name') OBJTYPE(*Q) USER(Bob) AUT(*GET) MQMNAME('MQ2')
GRTMQMAUT OBJ('SYSTEM.FTE.EVENT.agent_name') OBJTYPE(*Q) USER(Bob) AUT(*BROWSE) MQMNAME('MQ2')

GRTMQMAUT OBJ('SYSTEM.FTE.STATE.agent_name') OBJTYPE(*Q) USER(Bob) AUT(*PUT) MQMNAME('MQ2')
GRTMQMAUT OBJ('SYSTEM.FTE.STATE.agent_name') OBJTYPE(*Q) USER(Bob) AUT(*GET) MQMNAME('MQ2')
GRTMQMAUT OBJ('SYSTEM.FTE.STATE.agent_name') OBJTYPE(*Q) USER(Bob) AUT(*INQ) MQMNAME('MQ2')
```

z/OS For z/OS:

```
RDEFINE MQQUEUE MQ2.SYSTEM.FTE.REPLY.agent_name UACC(NONE)
PERMIT MQ2.SYSTEM.FTE.REPLY.agent_name CLASS(MQQUEUE) ID(Bob) ACCESS(UPDATE)

RDEFINE MQQUEUE MQ2.SYSTEM.FTE.DATA.agent_name UACC(NONE)
PERMIT MQ2.SYSTEM.FTE.DATA.agent_name CLASS(MQQUEUE) ID(Bob) ACCESS(UPDATE)

RDEFINE MQQUEUE MQ2.SYSTEM.FTE.EVENT.agent_name UACC(NONE)
PERMIT MQ2.SYSTEM.FTE.EVENT.agent_name CLASS(MQQUEUE) ID(Bob) ACCESS(UPDATE)
```

```
RDEFINE MQQUEUE MQ2.SYSTEM.FTE.STATE.agent_name UACC(NONE)
PERMIT MQ2.SYSTEM.FTE.STATE.agent_name CLASS(MQQUEUE) ID(Bob) ACCESS(UPDATE)
```

Authority to create a temporary reply queue for file transfers

Commands that connect to the command queue manager create a temporary queue on that queue manager. Once the command has submitted its request, it waits for a reply to come back on the temporary reply queue.

User Bob needs the `display`, `put`, `get`, and `browse` authorities on the temporary model queue definition.

For AIX, Linux, and Windows systems:

```
setmqaut -m MQ2 -t queue -n SYSTEM.DEFAULT.MODEL.QUEUE -m MQ2 -p Bob +put +dsp +get +browse
```

For IBM i:

```
GRTMQMAUT OBJ('SYSTEM.DEFAULT.MODEL.QUEUE') OBJTYPE(*Q) USER(Bob) AUT(*ADMDSP) MQMNAME('MQ2')
GRTMQMAUT OBJ('SYSTEM.DEFAULT.MODEL.QUEUE') OBJTYPE(*Q) USER(Bob) AUT(*PUT)    MQMNAME('MQ2')
GRTMQMAUT OBJ('SYSTEM.DEFAULT.MODEL.QUEUE') OBJTYPE(*Q) USER(Bob) AUT(*GET)    MQMNAME('MQ2')
GRTMQMAUT OBJ('SYSTEM.DEFAULT.MODEL.QUEUE') OBJTYPE(*Q) USER(Bob) AUT(*BROWSE) MQMNAME('MQ2')
```

For z/OS:

```
RDEFINE MQQUEUE MQ2.SYSTEM.DEFAULT.MODEL.QUEUE UACC(NONE)
PERMIT MQ2.SYSTEM.DEFAULT.MODEL.QUEUE CLASS(MQQUEUE) ID(Bob) ACCESS(UPDATE)
```

Authority to put messages on the SYSTEM.FTE queue and SYSTEM.FTE topic

`Put` permission is needed on the SYSTEM.FTE queue, and `publish` and `subscribe` permission on the SYSTEM.FTE topic, which are both hosted on the coordination queue manager.

For AIX, Linux, and Windows systems:

```
setmqaut -m MQ2 -t queue -n SYSTEM.FTE -m MQ2 -p Bob +put

setmqaut -m MQ2 -t topic -n SYSTEM.FTE -m MQ2 -p Bob +pub
setmqaut -m MQ2 -t topic -n SYSTEM.FTE -m MQ2 -p Bob +sub
```

For IBM i:

```
GRTMQMAUT OBJ('SYSTEM.FTE') OBJTYPE(*Q) USER(Bob) AUT(*PUT) MQMNAME('MQ2')

GRTMQMAUT OBJ('SYSTEM.FTE') OBJTYPE(*TOPIC) USER(Bob) AUT(*PUB) MQMNAME('MQ2')
GRTMQMAUT OBJ('SYSTEM.FTE') OBJTYPE(*TOPIC) USER(Bob) AUT(*SUB) MQMNAME('MQ2')
```

For z/OS:

```
RDEFINE MQQUEUE MQ2.SYSTEM.FTE UACC(NONE)
PERMIT MQ2.SYSTEM.FTE CLASS(MQQUEUE) ID(Bob) ACCESS(UPDATE)

RDEFINE MXTOPIC MQ2.PUBLISH.SYSTEM.FTE UACC(NONE)
PERMIT MQ2.PUBLISH.SYSTEM.FTE CLASS(MXTOPIC) ID(Bob) ACCESS(UPDATE)
```

Related reference

“Restricting group authorities for MFT-specific resources” on page 26

Instead of granting authority to individual users for all of the various objects that might be involved, configure two security groups for the purposes of administering Managed File Transfer access control: FTEUSER and FTEAGENT. It is the responsibility of the IBM MQ administrator to create and populate these groups. The administrator can choose to extend or modify the proposed configuration described here.

"Restricting user authorities on MFT agent actions" on page 31
In addition to using groups to manage access to resources, you can enable an additional level of security to restrict the Managed File Transfer agent actions that a user can take. Grant authorities on an agent authority queue to a user to give the user permission to perform specific agent actions.

Authorities for the MFT logger

Authority to publish MFT Agents log and status messages

Managed File Transfer Agents issue various log, progress, and status messages that are published on the coordination queue manager. The publication of these messages is subject to the IBM MQ security model, and in some cases you might have to perform further configuration to enable publication.

For more information about IBM MQ security, see the section starting with Securing IBM MQ.

Managed File Transfer agents flow messages for publication to the SYSTEM.FTE queue on the coordination queue manager. Each message carries a user ID in its message descriptor (MQMD). Messages are published using a topic object that is also called SYSTEM.FTE. For the publication of a given message to take place, the authority records of the SYSTEM.FTE topic must permit publication by the user ID contained in the MQMD of the message.

z/OS On z/OS, the channel initiator user ID needs access to publish to the SYSTEM.FTE topic. The user ID in the MQMD of the message also needs access to publish to this topic if the RESLEVEL security profile causes two user IDs to be checked for the channel initiator connection.

The user ID initially contained in the message depends on how the agent is connected to its own queue manager. Messages from bindings-connected agents contain the user ID that the agent is running under. Messages from client-connected agents contain an internal IBM MQ user ID.

You can change the user ID in a message. For both client- and bindings-connected agents, you can use the property publicationMDUser (in the `agent.properties` file) to specify a user ID, which is used in all log and status messages from that agent. The agent must be given permission by its own queue manager to use this alternative user ID; give this permission by granting setid authority to the user ID that the agent runs under.

You can also change the user ID contained in all messages from a client-connected agent using the MCAUSER property on the channel that the agent uses to connect to its queue manager.

You can change the user ID in messages using a channel exit, for example on the receiver channel bringing messages into the coordination queue manager.

Depending on the IBM MQ topology and policies, there are a number of ways an IBM MQ administrator can use the information in this topic to ensure that the publication of status and log messages takes place. Two examples are:

- Determine all the user IDs used by agents in the network. Explicitly grant an authority record for each of these IDs.
- Create one or more common user names to publish log and status messages. Create authority records for these user names on the coordination queue manager. Set the publicationMDUser property for each agent to a common user name. On each agent queue manager, grant setid authority to the user ID that the agent runs under to allow it to accept the publicationMDUser property.

Authorities for MFT to access file systems

For any file transfer request, the Managed File Transfer agent processes require some level of access to their local file systems.

- To transfer from a source file, the user ID that the source agent runs under must have read access to the source file. Additionally, you might need to give the source agent delete or write authority depending on the source disposition attribute.
- To transfer to a file or directory, the user ID that the destination agent runs under must have write authority to the specified path. Additionally, you might need to give the destination agent update authority, depending on the destination exists attribute.
- In addition to the file access authority that you grant to the agent process, you can also use sandboxing to specify and enforce a restricted file path area. For more information, see MFT sandboxes.
- If the files that you want to transfer to or from are not in a location accessible to the agent, for example z/OS a VSAM data set or in a location that is restricted by the sandboxing capability, you can use Managed File Transfer user exits to move the file to or from a location that can be accessed by the agent. For more information, see Customizing MFT with user exits.

commandPath MFT property

Use the **commandPath** property to specify the locations that Managed File Transfer can run commands from. Take extreme care when you set this property because any command in one of the specified commandPaths can effectively be called from a remote client system that is able to send commands to the agent.

You can specify a command to be run on the system where the agent is running from the managed transfer and managed call functions of Managed File Transfer. See Program invocation nested elements for information. However, commands must be on paths referenced by the **commandPath** agent property.

If the command specified is not fully qualified, Managed File Transfer attempts to find a matching command on the command path. If there is more than one matching command on the command path, the first match is used.

By default, the **commandPath** property is empty so that the agent cannot call any commands.

Specify the commandPath agent property as follows:

```
commandPath=command_directory_name
separator...command_directory_name
```

z/OS Or for z/OS only, specify:

```
commandPath=command_directory_name_or_data_set_name_prefix
separator...command_directory_name_or_data_set_name_prefix
```

where:

- *command_directory_name* is a directory path for commands that can be run.
- z/OS *command_directory_name_or_data_set_name_prefix* is a z/OS UNIX System Services directory path for commands that can be run, or a data set name prefix, that starts with //. You can choose to use a fully qualified or unqualified data set name prefix (that is, in the form: //'HLQ...' or //HLQ...). Specify partitioned data sets in the form //'HLQ()...' or //HLQ()... Use data sets to specify JCL script commands only.
- *separator* is the platform-specific separator.

Linux AIX For example, on AIX and Linux systems, if you want to run commands that are located in the directories `/home/user/cmds1` and `/home/user/cmds2`, set the **commandPath** agent property as follows:

```
commandPath=/home/user/cmds1:/home/user/cmds2
```

Parentheses, commas (,) and backslashes (\) are special characters in MFT commands, and must be escaped with a backslash (\) character. Windows File paths on Windows can be specified either using double backslashes (\\) as a separator, or using single forward slashes (/).

Windows For example, on a Windows system if you want to run commands that are located in the directories `C:\File Transfer\commands` and `C:\File Transfer\agent commands`, set the **commandPath** agent property as follows:

```
commandPath=C:\\File Transfer\\commands;C:\\File Transfer\\agent commands
```

z/OS For example, on z/OS if you want to run commands that are:

- In the directories `/home/user/cmds1` and `/home/user/cmds2`
- In data sets that start with `//'USER.CMD1', //CMD2,`
- Members of a fully qualified PDS named `//'USER.CMDS'`

set the commandPath agent property as follows:

```
commandPath=/home/user/cmds1:/home/user/cmds2://'USER.CMD1'://CMD2://'USER.CMDS()'
```

Important: Extreme care must be taken when you set this property, because any command in one of the specified commandPaths can be called from a remote client system that is able to send commands to the agent. For this reason, by default, when you specify a commandPath, sandboxing is configured so that all commandPath directories (and their subdirectories) are automatically denied access for a transfer:

- If the agent is configured to use an agent sandbox, the commandPath directories are automatically added to the list of denied directories when the agent starts.
- If the agent is configured with one or more user sandboxes, the commandPath directories are added as `<exclude>` elements to the `<read>` and `<write>` elements for each user sandbox when the agent starts up.
- If the agent is not configured to use either an agent sandbox, or user sandboxes, then a new agent sandbox is created when the agent starts up that has the commandPath directories specified as denied directories.

You can override this behavior by adding the following property to the `agent.properties` file:

```
addCommandPathToSandbox=false
```

When the `addCommandPathToSandbox` property is present and set to false, the following behavior occurs:

- If the agent is configured to use an agent sandbox, and the sandbox does not have any allowed directories specified, the commandPath directories are automatically added to the list of denied directories when the agent starts.
- If the agent is configured to use an agent sandbox, and the sandbox has one or more allowed directories specified, the commandPath directories are not added to the list of denied directories when the agent starts.
- If the agent is configured with one or more user sandboxes, the user sandboxes are not changed, and the commandPath directories are not added as `<exclude>` elements to the `<read>` and `<write>` elements for each user sandbox.

- If the agent is not configured to use either an agent sandbox, or user sandboxes, then a new agent sandbox is created when the agent starts up that has the commandPath directories specified as denied directories.

Related reference

The MFT `agent.properties` file

Cryptographic hardware

The way in which IBM MQ provides support for cryptographic hardware depends on which platform you are using.

ALW On AIX, Linux, and Windows systems, IBM MQ provides support for a variety of cryptographic hardware using the PKCS #11 interface.

z/OS IBM i On IBM i and z/OS, the operating system provides the cryptographic hardware support.

For a list of currently supported cryptography cards, see Cryptography Card List for IBM MQ.

On all platforms, cryptographic hardware is used at the TLS handshaking stage and at secret key reset.

IBM i On IBM i, when you use DCM to create or renew certificates, you can choose to store the key directly in the coprocessor or to use the coprocessor master key to encrypt the private key and store it in a special keystore file.

z/OS On z/OS, when you use RACF to create certificates, you can choose to store the key using ICSF (Integrated Cryptographic Service Facility) to obtain improved performance and more secure key storage. During the TLS handshake, and secret key negotiations, a crypto express card, (if available) is used to do RSA operations. After the handshake completes and data begins to flow, data is decrypted in the CPACF and the crypto express card is not used.

ALW On AIX, Linux, and Windows systems, IBM MQ support is also provided for TLS cryptographic hardware symmetric cipher operations. When using TLS cryptographic hardware symmetric cipher operations, data sent across a TLS connection is encrypted/decrypted by the cryptographic hardware product.

On the queue manager, this is enabled by setting the SSLCryptoHardware queue manager attribute appropriately (see ALTER QMGR and Change Queue Manager). On the WebSphere MQ MQI client, equivalent variables are provided (see SSL stanza of the client configuration file). The default setting is off.

If this attribute is enabled, IBM MQ attempts to use symmetric cipher operations whether the cryptographic hardware product supports them for the encryption algorithm specified in the current CipherSpec or not. If the cryptographic hardware product does not provide this support, IBM MQ performs the encryption and decryption of data itself, and no error is reported. If the cryptographic hardware product supports symmetric cipher operations for the encryption algorithm specified in the current CipherSpec, this function is activated and the cryptographic hardware product performs the encryption and decryption of the data sent.

In a situation of low processor usage it is often quicker to perform the encryption/decryption in software, rather than copying the data onto the card, encrypting/decrypting it, and copying it back to the TLS protocol software. Hardware symmetric cipher operations become more useful when the processor usage is high.

z/OS On z/OS with cryptographic hardware, support is provided for symmetric cipher operations. This means that the user's data is encrypted and decrypted by the hardware if the hardware has this capability for the CipherSpec chosen, and is configured to support data encryption and decryption.

IBM i On IBM i, cryptographic hardware is not used for encryption and decryption of the user's data, even if the hardware has the capability of performing such encryption for the encryption algorithm specified in the current CipherSpec.

IBM MQ rules for SSLPEER values

The **SSLPEER** attribute is used to check the Distinguished Name (DN) of the certificate from the peer queue manager or client at the other end of an IBM MQ channel. IBM MQ uses certain rules when comparing these values

Attention: The only peer values allowed for use in an **SSLPEER** filter are those shown in the table in Distinguished Names.

When **SSLPEER** values are compared with DNs, the rules for specifying and matching attribute values are as follows:

1. You can use either a comma or a semicolon as a separator.
2. Spaces before or after the separator are ignored. For example:

   ```
   CN=John Smith, O=IBM ,OU=Test , C=GB
   ```

3. The values of attribute types `SERIALNUMBER, MAIL, E, UID OR USERID, CN, T, OU, DC, O, STREET, L, ST, SP, S, PC, C, UNSTRUCTUREDNAME, UNSTRUCTUREDADDRESS, DNQ` are text strings that typically include only the following:

   - Uppercase and lowercase alphabetic characters A through Z and a through z
   - Numeric characters `0` through 9
   - The space character
   - Characters `, . ; ' " ( ) / -`

   To avoid conversion problems between different platforms, do not use other characters in an attribute value. The attribute types, for example `CN`, must be in uppercase characters.
4. Strings containing the same alphabetic characters match irrespective of case.
5. Spaces are not allowed between the attribute type and the = character.
6. Optionally, you can enclose attribute values in double quotation marks, for example `CN="John Smith"`. The quotation marks are discarded when matching values.
7. Spaces at either end of the string are ignored unless the string is enclosed in double quotation marks.
8. The comma and semicolon attribute separator characters are considered to be part of the string when enclosed in double quotation marks, or when a backslash escape character (\) precedes them.
9. The names of attribute types, for example `CN` or `OU`, are considered to be part of the string when enclosed in double quotation marks.
10. Any of the attribute types ST, SP, and S can be used for the State or Province name.
11. Any attribute value can have an asterisk (`*`) as a pattern-matching character at the beginning, the end, or in both places. The asterisk character substitutes for any number of characters at the beginning or end of the string to be matched. This character enables your SSLPEER value specification to match a range of Distinguished Names. For example, `OU=IBM*` matches every Organizational Unit beginning with IBM, such as IBM Corporation.

    The asterisk character can also be a valid character in a Distinguished Name. To obtain an exact match with an asterisk at the beginning or end of the string, the backslash escape character (\) must precede the asterisk: `\*`. Asterisks in the middle of the string are considered to be part of the string and do not require the backslash escape character.
12. The DN can contain multiple OU attributes and multiple DC attributes.
13. When multiple OU attributes are specified, all must exist and be in descending hierarchical order. For an example, see DEFINE CHANNEL.

14. A digital certificate Subject DN can additionally contain multiple attributes of the same type other than OU or DC, but only if the **SSLPEER** value does not filter on the repeated attribute type. For example, consider a certificate with the following Subject DN:

    ```
    CN=First, CN=Second, O=IBM, C=US
    ```

    An **SSLPEER** value of `O=IBM, C=US` does not filter on CN, so matches this certificate and allows the connection. An **SSLPEER** value of `CN=First, O=IBM, C=US` fails to match this certificate because the certificate contains multiple CN attributes. You cannot match multiple CN values.

15. The value of the **SERIALNUMBER** attribute should consist of colon separated bytes as hexadecimal, for example `A1:B2:C3`.

Related concepts

Distinguished Names

Channel authentication records

Related tasks

Mapping a TLS Distinguished Name to an MCAUSER user ID

Multi GSKit: Digital certificate signature algorithms compliant with FIPS 140-3

The list of digital certificate signature algorithms in IBM Global Security Kit (GSKit) that are compliant with FIPS 140-3

Notes:

- V 10.0.0 V 10.0.0 ALW On AIX, Linux, and Windows, IBM MQ provides FIPS 140-3 compliance through the GSKit 9 IBM Crypto for C (ICC) cryptographic module. The NIST certification associated with the FIPS 140-3 module can be viewed at https://csrc.nist.gov/projects/cryptographic-module-validation-program/certificate/4755.
- V 10.0.0 V 10.0.0 FIPS support is currently not available for Linux s390x. Customers on this platform who require FIPS support should remain at a previous version of IBM MQ. Support for FIPS 140-3 will be enabled in a future FixPack once the IBM Crypto for C (ICC) cryptographic module has received its certification on this platform.
- V 10.0.0 The FIPS 140-3 cryptographic module within IBM Semeru Runtime was approved by NIST in August 2024. IBM MQ 10.0.0 adds support for the handling of IBM MQ classes for JMS and IBM MQ classes for Java client connections using TLS for FIPS 140-3 in Java 8 and IBM Semeru Runtime 11+. The NIST certification associated with the FIPS 140-3 module can be viewed at https://csrc.nist.gov/projects/cryptographic-module-validation-program/certificate/4755.
- For IBM MQ in Containers, the IBM MQ Operator 3.2.0 and queue manager container image 9.4.0.0 onwards are based on UBI 9. FIPS 140-3 compliance for IBM MQ in Containers is currently pending.

  V 10.0.0 If FIPS is enabled, IBM MQ in Container control processes use a FIPS 140-3 Certified OpenSSL Module. Details of the NIST certification can be viewed at: https://access.redhat.com/compliance/fips. IBM MQ queue managers running in container images have the same FIPS certification level as the base image platform version of IBM MQ.

- RSA with SHA-224
- RSA with SHA-256
- RSA with SHA-384
- RSA with SHA-512
- ECDSA with SHA-224
- ECDSA with SHA-256
- ECDSA with SHA-384
- ECDSA with SHA-512

- Curve P-224
- Curve P-256
- Curve P-384
- Curve P-521
- Curve K-233
- Curve K-283
- Curve K-409
- Curve K-571
- Curve B-233
- Curve B-283
- Curve B-409
- Curve B-571

Related concepts
Digital certificates and CipherSpec compatibility in IBM MQ

GSKit return codes used in AMS messages

This topic describes the IBM Global Security Kit (GSKit) return codes that appear in some Advanced Message Security (AMS) messages.

If you receive a numeric return code generated by GSKit, refer to the following tables to determine the message code or the explanation.

In most cases the reason code listed in Table 2 will be more specific to your error and thus assist you with diagnosing the issue.

Table 7. GSKit error messages insert sorted by decimal error code

Decimal Error Code	Message Code	Explanation
0	GSS_S_COMPLETE	The operation was successful. There is not an error.
65536	GSS_S_BAD_MECH	The keystore type does not support the requested mechanism.
131072	GSS_S_BAD_NAME	Internal error.
196608	GSS_S_BAD_NAMETYPE	Internal error.
262144	GSS_S_BAD_BINDINGS	Internal error.
327680	GSS_S_BAD_STATUS	Internal error.
393216	GSS_S_BAD_SIG	The signature is incorrect.
458752	GSS_S_NO_CRED	There is no valid keystore type.
524288	GSS_S_NO_CONTEXT	Internal error.
589824	GSS_S_DEFECTIVE_TOKEN	Internal error.
655360	GSS_S_DEFECTIVE_CREDENTIAL	Internal error.
720896	GSS_S_CREDENTIALS_EXPIRED	Internal error.
786432	GSS_S_CONTEXT_EXPIRED	Internal error.

Table 7. GSKit error messages insert sorted by decimal error code (continued)		
Decimal Error Code	**Message Code**	**Explanation**
851968	GSS_S_FAILURE	The operation failed. Examine the minor reason code for more details.
917504	GSS_S_BAD_QOP	An invalid protection policy was provided.
983040	GSS_S_UNAUTHORIZED	Unable to protect or remove protection due to permissions.
1048576	GSS_S_UNAVAILABLE	Internal error.
1114112	IDUP_S_BAD_TARG_INFO	Internal error.
1179648	IDUP_S_BAD_DOA_KEY	Internal error.
1245184	IDUP_S_BAD_KE_KEY	Internal error.
1310720	IDUP_S_BAD_ENC_IDU	Internal error.
1376256	IDUP_S_EVIDENCE_TOKEN_INCOMPLETE	Internal error.
1441792	IDUP_S_SERV_VERIF_INFO_NEEDED	Internal error.
1507328	IDUP_S_SERVICE_UNAVAIL	Internal error.
1572864	IDUP_S_REQ_TIME_SERVICE_UNAVAIL	Internal error.
1638400	IDUP_S_NO_ENV	Internal error.
1703936	IDUP_S_NO_MATCH	Internal error.
1769472	IDUP_S_UNKNOWN_OPER_ID	Internal error.
1835008	GSS_S_NAME_NOT_MN	Internal error.
2621440	GSS_S_SMIME_NOT_INITIALIZED	Internal error.
2686976	GSS_S_SMIME_NO_USER_DATA	Internal error.
2752512	GSS_S_SMIME_INVALID_SMIME_MESSAGE	Internal error.
2818048	GSS_S_SMIME_NOT_LOGGED_IN	Internal error.

Table 8. GSKit error messages sorted by decimal reason code		
Decimal Reason Code	**Message Code**	**Explanation**
0	GSS_S_MINOR_OK	OK. There is not an error.
0	GSS_S_MINOR_SUCCESS	OK. There is not an error
1	GSS_S_MINOR_MEMORY_ALLOCATION_FAILURE	A general purpose memory allocation failure has occurred.
1	GSS_S_MINOR_INSUFFICIENT_STORAGE	A general purpose memory allocation failure has occurred.

Table 8. GSKit error messages sorted by decimal reason code (continued)		
Decimal Reason Code	**Message Code**	**Explanation**
2	GSS_S_MINOR_NOT_MECHANISM_NAME	The name is not a mechanism name.
3	GSS_S_MINOR_INVALID_NAME	The provided name is invalid.
4	GSS_S_MINOR_GSK_ERROR	GSKit has returned an error.
5	GSS_S_MINOR_NO_MORE_NAME	There are no more names to parse from the name object.
6	GSS_S_MINOR_MEMBER_NOT_FOUND	An object was referenced from a set, but the requested object could not be found.
7	GSS_S_MINOR_BAD_QUALITY_OF_PROTECTION_ALGORITHM	The Quality of Protection algorithm is bad.
8	GSS_S_MINOR_BAD_QUALITY_OF_SIGNING_ALGORITHM	The Quality of Signing algorithm is bad.
9	GSS_S_MINOR_BAD_DIGEST_ENCRYPTION_ALGORITHM	The digest encryption algorithm is bad.
10	GSS_S_MINOR_BAD_INPUT	One or more required input parameters is NULL.
11	GSS_S_MINOR_HANDLE_INVALID	The object handle is invalid.
12	GSS_S_MINOR_NO_PRIVKEY_IN_KEYRING	There is no entry with a private key in the database.
12	GSS_S_MINOR_NO_PRIVKEY_IN_DB	There is no entry with a private key in the database.
13	GSS_S_MINOR_BAD_KEYRING_TYPE	The database entry type is bad.
14	GSS_S_MINOR_KEYRING_ACCESS_EXCEPTION	An exception in accessing the database has occurred. Additional information: ensure all GSKit libraries can be accessed and are not corrupted.
15	GSS_S_MINOR_API_NOT_SUPPORTED	The Application Interface (API) is not supported.
16	GSS_S_MINOR_CREDENTIAL_STILL_EXISTS	The credential still exists.
17	GSS_S_MINOR_ENV_STILL_EXISTS	The environment still exists.
18	GSS_S_MINOR_EXPIRED_CREDENTIAL	The credential has expired.
19	GSS_S_MINOR_NO_SIGNER	No signer is available for the specified credential.
20	GSS_S_MINOR_PIDU_HAD_INVALID_CONTENT_TYPE	The protected independent data unit (PIDU) has an invalid content type.

Table 8. GSKit error messages sorted by decimal reason code (continued)

Decimal Reason Code	Message Code	Explanation
21	GSS_S_MINOR_PIDU_HAD_INVALID_CONTENT_ENCRYPTION_ALGORITHM	The protected independent data unit (PIDU) has an invalid content encryption algorithm.
22	GSS_S_MINOR_BLOB_ALREADY_EXISTS	The blob already exists.
23	GSS_S_MINOR_INVALID_MECH	A mechanism-type object identifier is syntactically invalid.
24	GSS_S_MINOR_MECH_NOT_SUPPORTED	The indicated mechanism type is not supported in this implementation.
25	GSS_S_MINOR_STATIC_OID	There is an attempt to free an object identifier (OID) which is static and cannot be freed.
26	GSS_S_MINOR_PIDU_INVALID_SESSION_KEY	The protected independent data unit (PIDU) has a session key that cannot be used to decrypt the data.
27	GSS_S_MINOR_PIDU_RECIPIENT_INFO_INVALID	The protected independent data unit (PIDU) refers to a certificate that cannot be used to decrypt the session key.
28	GSS_S_MINOR_PIDU_HAS_UNSUPPORTED_DIGEST_ALGORITHM	The protected independent data unit (PIDU) has an unsupported digest algorithm.
29	GSS_S_MINOR_PIDU_HAS_UNSUPPORTED_DIGEST_ENCRYPTION	The protected independent data unit (PIDU) has an unsupported digest encryption algorithm.
30	GSS_S_MINOR_SIGNING_NOT_ALLOWED_BY_ENV	The environment is not set up to do a signing operation. Additional information: the key usage bits in the certificate might not allow the specified operation.
31	GSS_S_MINOR_ENCRYPTION_NOT_ALLOWED_BY_ENV	The environment is not set up to do an encryption operation. Additional information: the key usage bits in the certificate might not allow the specified operation.
32	GSS_S_MINOR_NO_VALID_TARGET_NAMES_IN_DATABASE	None of the specified names were found in the database.
33	GSS_S_MINOR_NO_VALID_SIGNERS	No signers could be validated while unprotecting a signed protected independent data unit (PIDU).

Table 8. GSKit error messages sorted by decimal reason code (continued)		
Decimal Reason Code	**Message Code**	**Explanation**
34	GSS_S_MINOR_MULTIPLE_SIGNERS	There are multiple signers in the signed protected independent data unit (PIDU); however, only the first one is being returned.
35	GSS_S_MINOR_BAD_SEQUENCE	The multi-buffer has been called out of order (For example, end_unprotect is called after start_protect).
36	GSS_S_MINOR_INVALID_NAMETYPE	The nametype argument provided is invalid.
37	GSS_S_MINOR_FAILURE	A general internal failure has occurred.
38	GSS_S_MINOR_BAD_OID	The object identifier provided is syntactically invalid.
39	GSS_S_MINOR_INVALID_CREDENTIAL	The credential is invalid.
40	GSS_S_MINOR_INVALID_ENVIRONMENT	The environment is invalid.
41	GSS_S_MINOR_VERIFY_NOT_ALLOWED_BY_ENV	The environment is not set up to verify the operation. The environment is not set up to verify the operation. Additional information: the key usage bits in the certificate might not allow the specified operation.
42	GSS_S_MINOR_DECRYPTION_NOT_ALLOWED_BY_ENV	The environment is not set up to do a decryption operation. Additional information: the key usage bits in the certificate might not allow the specified operation.
43	GSS_S_MINOR_UNABLE_TO_DECRYPT_PIDU	The protected independent data unit (PIDU) cannot be decrypted. Additional information: ensure the recipients extended attribute on the privacy-protected queue includes the certificate DN of the actual recipient of the message. Additionally, ensure that the public key that the sender has for the recipient DN matches the private key in the recipient's keystore.
44	GSS_S_MINOR_INVALID_PKCS7_MESSAGE	An Invalid PKCS7 message has been received.

Table 8. GSKit error messages sorted by decimal reason code (continued)

Decimal Reason Code	Message Code	Explanation
45	GSS_S_MINOR_USAGE_VALIDATION_FAILED	The application was not built with the right level of GSKit/ACME or is not permitted to use ACME API interface.
46	GSS_S_MINOR_DIGEST_ERROR	An error occurred during the message digest and the message is possibly corrupted.
47	GSS_S_MINOR_ENCRYPTION_ERROR	An error occurred during the data encryption and the message is possibly corrupted.
48	GSS_S_MINOR_DECRYPTION_ERROR	An error occurred during the data decryption and the message is possibly corrupted.
49	GSS_S_MINOR_ACCELERATOR_NOT_SUPPORTED	The specified card is either not supported or has not been installed properly.
50	GSS_S_MINOR_PKCS11_TOKEN_NOTPRESENT	The PKCS #11 token could not be found.
51	GSS_S_MINOR_PKCS11_TOKEN__LABEL_MISMATCH	The PKCS #11 token label was not entered correctly.
52	GSS_S_MINOR_PKCS11_TOKEN_INVALID_PIN	The user PIN entered for the PKCS #11 token is invalid.
53	GSS_S_MINOR_PKCS11_LIBRARY_NOT_LOADED	The system could not load the PKCS #11 library.
54	GSS_S_MINOR_DECODING_ERROR	An error occurred during Base 64 or ASN.1 decoding for either the certificate or distinguished name.
55	GSS_S_MINOR_SIGN_ERROR	An error occurred during the signing process.
56	GSS_S_MINOR_VERIFY_ERROR	An error occurred during the signature verification process.
57	GSS_S_MINOR_RECIPIENT_CERT_NOT_FOUND	The application could not locate the recipient certificate.
58	GSS_S_MINOR_CERT_HpAS_NO_PRIVATE_KEY	The certificate does not have a private encryption key.
59	GSS_S_MINOR_CERT_HAS_BAD_VALIDITY_DATE	The certificate has a wrong validity date.
60	GSS_S_MINOR_BAD_CERTIFICATE	The certificate is not valid.
61	GSS_S_MINOR_FIPS_NOT_SUPPORTED	FIPS mode is not supported in this version.
62	GSS_S_MINOR_SIGNER_CERT_BAD	The signer certificate is not trusted.

Table 8. GSKit error messages sorted by decimal reason code (continued)		
Decimal Reason Code	**Message Code**	**Explanation**
63	GSS_S_MINOR_SIGNER_CERT_BAD_DATE	The signer certificate has a bad validity date.

Monitoring reference

Use the reference information in this section to help you monitor IBM MQ.

- “Structure data types” on page 50
- “Object attributes for event data” on page 74
- “Event message reference” on page 119

Related tasks

Monitoring and performance

Structure data types

Use this topic to understand the structure data types used in the message data that IBM MQ monitoring techniques generate.

The subtopics describe in a language-independent form the structure data types used in monitor message data.

- “MQCFBS - Byte string parameter” on page 50
- “MQCFGR - Group parameter” on page 52
- “MQCFH - PCF header” on page 54
- “MQCFIL - Integer list parameter” on page 58
- “MQCFIL64 - 64-bit integer list parameter” on page 60
- “MQCFIN - Integer parameter” on page 62
- “MQCFIN64 - 64-bit integer parameter” on page 64
- “MQCFSL - String list parameter” on page 65
- “MQCFST - String parameter” on page 68
- “MQEPH - Embedded PCF header” on page 70

The declarations are shown in the following programming languages:

- C
- COBOL
- PL/I
- IBM i RPG (ILE) (IBM i only)
- z/OS S/390 assembler (z/OS only)
- Windows Visual Basic (Windows only)

MQCFBS - Byte string parameter

Use this page to view the structure of an MQCFBS parameter and the declarations for the following programming languages: C, COBOL, PL/I, RPG/ILE, and S/390 assembler

The MQCFBS structure describes a byte string parameter. Following the links to the declarations is a description of the fields making up the MQCFBS structure:

- C language

- COBOL language
- PL/I language (z/OS only)
- RPG/ILE language (IBM i only)
- S/390 assembler-language (z/OS only)

Type

Description: This indicates that the structure is an MQCFBS structure describing a byte string parameter.

Data type: MQLONG.

Value:

MQCFT_BYTE_STRING
Structure defining a byte string.

StrucLength

Description: This is the length in bytes of the MQCFBS structure, including the variable-length string at the end of the structure (the *String* field).

Data type: MQLONG.

Parameter

Description: This identifies the parameter with a value that is contained in the structure.

Data type: MQLONG.

StringLength

Description: This is the length in bytes of the data in the *String* field, and is zero or greater.

Data type: MQLONG.

String

Description: This is the value of the parameter identified by the *Parameter* field. The string is a byte string, and so is not subject to character-set conversion when sent between different systems.

Note: A null byte in the string is treated as normal data, and does not act as a delimiter for the string.

Data type: MQBYTE x *StringLength*.

C language declaration

```
struct tagMQCFBS {
  MQLONG  Type;          /* Structure type */
  MQLONG  StrucLength;   /* Structure length */
  MQLONG  Parameter;     /* Parameter identifier */
  MQLONG  StringLength;  /* Length of string */
  MQBYTE  String[1];     /* String value -- first character */
 } MQCFBS;
```

COBOL language declaration

```
**   MQCFBS structure
  10 MQCFBS.
```

```
**    Structure type
   15 MQCFBS-TYPE         PIC S9(9) BINARY.
**    Structure length
   15 MQCFBS-STRUCLENGTH  PIC S9(9) BINARY.
**    Parameter identifier
   15 MQCFBS-PARAMETER    PIC S9(9) BINARY.
**    Length of string
   15 MQCFBS-STRINGLENGTH PIC S9(9) BINARY.
```

PL/I language declaration (z/OS only)

```
dcl
 1 MQCFBS based,
  3 Type         fixed bin(31), /* Structure type */
  3 StrucLength  fixed bin(31), /* Structure length */
  3 Parameter    fixed bin(31), /* Parameter identifier */
  3 StringLength fixed bin(31); /* Length of string */
```

RPG/ILE language declaration (IBM i only)

```
     D*..1....:....2....:....3....:....4....:....5....:....6....:....7..
     D* MQCFBS Structure
     D*
     D* Structure type
     D  BSTYP                  1      4I 0 INZ(9)
     D* Structure length
     D  BSLEN                  5      8I 0 INZ(16)
     D* Parameter identifier
     D  BSPRM                  9     12I 0 INZ(0)
     D* Length of string
     D  BSSTL                 13     16I 0 INZ(0)
     D* String value -- first byte
     D  BSSRA                 17     17    INZ
```

S/390 assembler-language declaration (z/OS only)

```
MQCFBS               DSECT
MQCFBS_TYPE          DS   F  Structure type
MQCFBS_STRUCLENGTH   DS   F  Structure length
MQCFBS_PARAMETER     DS   F  Parameter identifier
MQCFBS_STRINGLENGTH  DS   F  Length of string
*
MQCFBS_LENGTH        EQU  *-MQCFBS
                     ORG  MQCFBS
MQCFBS_AREA          DS   CL(MQCFBS_LENGTH)
```

MQCFGR - Group parameter

Use this page to view the structure of an MQCFGR parameter and the declarations for the following programming languages: C, COBOL, PL/I, RPG/ILE, S/390 assembler, and Visual Basic

The MQCFGR structure describes a group parameter. Following the links to the declarations is a description of the fields making up the MQCFGR structure:

- C language
- COBOL language
- PL/I language (z/OS only)
- RPG/ILE language (IBM i only)
- System/390 assembler-language (z/OS only)
- Visual Basic language (Windows only)

The MQCFGR structure is a group parameter in which the subsequent parameter structures are grouped together as a single logical unit. The number of subsequent structures that are included is given by *ParameterCount*. This structure, and the parameter structures it includes, are counted as one structure only in the *ParameterCount* parameter in the PCF header (MQCFH) and the group parameter (MQCFGR).

Type

Description: Indicates that the structure type is MQCFGR describing which parameters are in this group.

Data type: MQLONG.

Value:

MQCFT_GROUP
Structure defining a group of parameters.

StrucLength

Description: Length in bytes of the MQCFGR structure.

Data type: MQLONG.

Value:

MQCFGR_STRUC_LENGTH
Length of the command format group-parameter structure.

Parameter

Description: This identifies the type of group parameter.

Data type: MQLONG.

ParameterCount

Description: The number of parameter structures following the MQCFGR structure that are contained within the group identified by the *Parameter* field. If the group itself contains one or more groups, each group and its parameters count as one structure only.

Data type: MQLONG.

C language declaration

```
typedef struct tagMQCFGR {
  MQLONG  Type;            /* Structure type */
  MQLONG  StrucLength;     /* Structure length */
  MQLONG  Parameter;       /* Parameter identifier */
  MQLONG  ParameterCount;  /* Count of the grouped parameter structures */
 } MQCFGR;
```

COBOL language declaration

```
**   MQCFGR structure
  10 MQCFGR.
**    Structure type
   15 MQCFGR-TYPE           PIC S9(9) BINARY.
**    Structure length
   15 MQCFGR-STRUCLENGTH    PIC S9(9) BINARY.
**    Parameter identifier
   15 MQCFGR-PARAMETER      PIC S9(9) BINARY.
**    Count of grouped parameter structures
   15 MQCFGR-PARAMETERCOUNT PIC S9(9) BINARY.
```

PL/I language declaration (z/OS and Windows only)

```
dcl
 1 MQCFGR based,
  3 Type           fixed bin(31), /* Structure type */
  3 StrucLength    fixed bin(31), /* Structure length */
  3 Parameter      fixed bin(31), /* Parameter identifier */
  3 ParameterCount fixed bin(31), /* Count of grouped parameter structures */
```

RPG/ILE declaration (IBM i only)

```
     D*..1....:....2....:....3....:....4....:....5....:....6....:....7..
        D* MQCFGR Structure
     D*
     D* Structure type
     D  GRTYP                  1      4I INZ(20)
     D* Structure length
     D  GRLEN                  5      8I INZ(16)
     D* Parameter identifier
     D  GRPRM                  9     12I INZ(0)
     D* Count of grouped parameter structures
     D  GRCNT                 13     16I INZ(0)
     D*
```

S/390 assembler-language declaration (z/OS only)

```
MQCFGR                         DSECT
MQCFGR_TYPE                    DS   F        Structure type
MQCFGR_STRUCLENGTH             DS   F        Structure length
MQCFGR_PARAMETER               DS   F        Parameter identifier
MQCFGR_PARAMETERCOUNT          DS   F        Count of grouped parameter structures
MQCFGR_LENGTH                  EQU  *-MQCFGR Length of structure
                               ORG  MQCFGR
MQCFGR_AREA                    DS   CL(MQCFGR_LENGTH)
```

Visual Basic language declaration (Windows only)

```
Type MQCFGR
  Type As Long            ' Structure type
  StrucLength As Long     ' Structure length
  Parameter As Long       ' Parameter identifier
  ParameterCount As Long  ' Count of grouped parameter structures
  End Type
```

MQCFH - PCF header

Use this page to view the structure of an MQCFH header and the declarations for the following programming languages: C, COBOL, PL/I, RPG/ILE, S/390 assembler, and Visual Basic

The MQCFH structure describes the information that is present at the start of the message data of a monitoring message. Following the links to the declarations is a description of the fields making up the MQCFH structure:

- C language
- COBOL language
- PL/I language (z/OS only)
- RPG/ILE language (IBM i only)
- S/390 assembler language (z/OS only)
- Visual Basic language (Windows only)

Type

Description:	Structure type This indicates the content of the message.
Data type:	MQLONG.

Values:

MQCFT_ACCOUNTING
Message is an accounting message.

MQCFT_EVENT
Message is reporting an event.

MQCFT_REPORT
Message is an activity report.

MQCFT_RESPONSE
Message is a response to a command.

MQCFT_STATISTICS
Message is a statistics message.

MQCFT_TRACE_ROUTE
Message is a trace-route message.

StrucLength

Description: This is the length in bytes of the MQCFH structure

Data type: MQLONG.

Value:

MQCFH_STRUC_LENGTH
Length of command format header structure.

Version

Description: Structure version number.

Data type: MQLONG.

Value:

MQCFH_VERSION_1
Version number for all events except configuration and command events.

MQCFH_VERSION_2
Version number for configuration events.

MQCFH_VERSION_3
Version number for command events, activity reports, trace-route messages, accounting and statistics messages.

Command

Description: Specifies the category of the message.

Data type: MQLONG.

Value: Refer to the *Command* values in the following structure descriptions:

- "Event message MQCFH (PCF header)" on page 125.
- Activity report MQCFH (PCF header).
- Trace-route message MQCFH (PCF header).
- Message data in accounting and statistics messages.

MsgSeqNumber

Description: Message sequence number. This is the sequence number of the message within a set of related messages.

Data type: MQLONG.

Control

Description: Control options.

Data type: MQLONG.

Value:

MQCFC_LAST
Last message in the set.

MQCFC_NOT_LAST
Not the last message in the set.

CompCode

Description: Completion code.

Data type: MQLONG.

Value:

MQCC_OK
Events reporting OK condition, activity reports, trace-route messages, accounting messages, or statistics messages.

MQCC_WARNING
Event reporting warning condition.

Reason

Description: Reason code qualifying completion code.

Data type: MQLONG.

Value: For event messages:

MQRC_*
Dependent on the event being reported.

Note: Events with the same reason code are further identified by the **ReasonQualifier** parameter in the event data.

For activity reports, trace-route messages, accounting messages, and statistics messages:

MQRC_NONE

ParameterCount

Description: Count of parameter structures. This is the number of parameter structures that follow the MQCFH structure.

Data type: MQLONG.

Value: 0 or greater.

C language declaration

```
typedef struct tagMQCFH {
  MQLONG  Type;            /* Structure type */
  MQLONG  StrucLength;     /* Structure length */
  MQLONG  Version;         /* Structure version number */
  MQLONG  Command;         /* Command identifier */
  MQLONG  MsgSeqNumber;    /* Message sequence number */
  MQLONG  Control;         /* Control options */
  MQLONG  CompCode;        /* Completion code */
  MQLONG  Reason;          /* Reason code qualifying completion code */
  MQLONG  ParameterCount;  /* Count of parameter structures */
 } MQCFH;
```

COBOL language declaration

```
**   MQCFH structure
  10 MQCFH.
**    Structure type
   15 MQCFH-TYPE           PIC S9(9) BINARY.
**    Structure length
   15 MQCFH-STRUCLENGTH    PIC S9(9) BINARY.
**    Structure version number
   15 MQCFH-VERSION        PIC S9(9) BINARY.
**    Command identifier
   15 MQCFH-COMMAND        PIC S9(9) BINARY.
**    Message sequence number
   15 MQCFH-MSGSEQNUMBER   PIC S9(9) BINARY.
**    Control options
   15 MQCFH-CONTROL        PIC S9(9) BINARY.
**    Completion code
   15 MQCFH-COMPCODE       PIC S9(9) BINARY.
**    Reason code qualifying completion code
   15 MQCFH-REASON         PIC S9(9) BINARY.
**    Count of parameter structures
   15 MQCFH-PARAMETERCOUNT PIC S9(9) BINARY.
```

PL/I language declaration (z/OS and Windows)

```
dcl
 1 MQCFH based,
  3 Type           fixed bin(31), /* Structure type */
  3 StrucLength    fixed bin(31), /* Structure length */
  3 Version        fixed bin(31), /* Structure version number */
  3 Command        fixed bin(31), /* Command identifier */
  3 MsgSeqNumber   fixed bin(31), /* Message sequence number */
  3 Control        fixed bin(31), /* Control options */
  3 CompCode       fixed bin(31), /* Completion code */
  3 Reason         fixed bin(31), /* Reason code qualifying completion
                                     code */
  3 ParameterCount fixed bin(31); /* Count of parameter structures */
```

RPG language declaration (IBM i only)

```
D*..1....:....2....:....3....:....4....:....5....:....6....:....7..
D* MQCFH Structure
D*
D* Structure type
D  FHTYP                  1      4I 0 INZ(1)
D* Structure length
D  FHLEN                  5      8I 0 INZ(36)
D* Structure version number
D  FHVER                  9     12I 0 INZ(1)
D* Command identifier
D  FHCMD                 13     16I 0 INZ(0)
D* Message sequence number
D  FHSEQ                 17     20I 0 INZ(1)
D* Control options
D  FHCTL                 21     24I 0 INZ(1)
D* Completion code
D  FHCMP                 25     28I 0 INZ(0)
D* Reason code qualifying completion code
D  FHREA                 29     32I 0 INZ(0)
D* Count of parameter structures
D  FHCNT                 33     36I 0 INZ(0)
D*
```

S/390 assembler language declaration (z/OS only)

```
MQCFH                          DSECT
MQCFH_TYPE                     DS   F        Structure type
MQCFH_STRUCLENGTH              DS   F        Structure length
MQCFH_VERSION                  DS   F        Structure version number
MQCFH_COMMAND                  DS   F        Command identifier
MQCFH_MSGSEQNUMBER             DS   F        Message sequence number
MQCFH_CONTROL                  DS   F        Control options
MQCFH_COMPCODE                 DS   F        Completion code
```

```
MQCFH_REASON                    DS   F        Reason code qualifying
*                                            completion code
MQCFH_PARAMETERCOUNT            DS   F        Count of parameter
*                                            structures
MQCFH_LENGTH                    EQU  *-MQCFH  Length of structure
                                ORG  MQCFH
MQCFH_AREA                      DS   CL(MQCFH_LENGTH)
```

Visual Basic language declaration (Windows only)

```
Type MQCFH
  Type As Long            'Structure type
  StrucLength As Long     'Structure length
  Version As Long         'Structure version number
  Command As Long         'Command identifier
  MsgSeqNumber As Long    'Message sequence number
  Control As Long         'Control options
  CompCode As Long        'Completion code
  Reason As Long          'Reason code qualifying completion code
  ParameterCount As Long  'Count of parameter structures
End Type
```

MQCFIL - Integer list parameter

Use this page to view the structure of an MQCFIL parameter and the declarations for the following programming languages: C, COBOL, PL/I, RPG/ILE, S/390 assembler, and Visual Basic

The MQCFIL structure describes an integer list parameter. Following the links to the declarations is a description of the fields making up the MQCFIL structure:

- C language
- COBOL language
- PL/I language (z/OS only)
- RPG/ILE language (IBM i only)
- System/390 assembler-language (z/OS only)
- Visual Basic language (Windows only)

Type

Description: Indicates that the structure type is MQCFIL and describes an integer-list parameter.

Data type : MQLONG.

Value:

MQCFT_INTEGER_LIST
Structure defining an integer list.

StrucLength

Description: Length in bytes of the MQCFIL structure, including the array of integers at the end of the structure (the *values* field).

Data type : MQLONG.

Parameter

Description: Identifies the parameter with a value that is contained in the structure.

Data type : MQLONG.

Count

Description: Number of elements in the *Values* array.

Data type : MQLONG.

Values: Zero or greater.

Values

Description: Array of values for the parameter identified by the *Parameter* field.

Data type : MQLONG x *Count*

The way that this field is declared depends on the programming language:

- For the C programming language, the field is declared as an array with one element. Storage for the structure must be allocated dynamically, and pointers used to address the fields within it.
- For the COBOL, PL/I, RPG, and System/390 assembler programming languages, the field is omitted from the structure declaration. When an instance of the structure is declared, you must include MQCFIL in a larger structure, and declare additional fields following MQCFIL, to represent the Values field as required.

C language declaration

```
typedef struct tagMQCFIL {
  MQLONG  Type;         /* Structure type */
  MQLONG  StrucLength;  /* Structure length */
  MQLONG  Parameter;    /* Parameter identifier */
  MQLONG  Count;        /* Count of parameter values */
  MQLONG  Values[1];    /* Parameter values - first element */
 } MQCFIL;
```

COBOL language declaration

```
**   MQCFIL structure
  10 MQCFIL.
**    Structure type
   15 MQCFIL-TYPE        PIC S9(9) BINARY.
**    Structure length
   15 MQCFIL-STRUCLENGTH PIC S9(9) BINARY.
**    Parameter identifier
   15 MQCFIL-PARAMETER   PIC S9(9) BINARY.
**    Count of parameter values
   15 MQCFIL-COUNT       PIC S9(9) BINARY.
```

PL/I language declaration

```
dcl
 1 MQCFIL based,
  3 Type        fixed bin(31), /* Structure type */
  3 StrucLength fixed bin(31), /* Structure length */
  3 Parameter   fixed bin(31), /* Parameter identifier */
  3 Count       fixed bin(31); /* Count of parameter values */
```

RPG/ILE declaration (IBM i only)

```
     D*..1....:....2....:....3....:....4....:....5....:....6....:....7..
     D* MQCFIL Structure
     D*
     D* Structure type
     D  ILTYP                  1      4I 0
     D* Structure length
     D  ILLEN                  5      8I 0
     D* Parameter identifier
     D  ILPRM                  9     12I 0
     D* Count of paramter valuee
     D  ILCNT                 13     16I 0
```

S/390 assembler-language declaration

```
MQCFIL                          DSECT
MQCFIL_TYPE                     DS   F        Structure type
MQCFIL_STRUCLENGTH              DS   F        Structure length
MQCFIL_PARAMETER                DS   F        Parameter identifier
MQCFIL_COUNT                    DS   F        Count of parameter values
MQCFIL_LENGTH                   EQU  *-MQCFIL Length of structure
                                ORG  MQCFIL
MQCFIL_AREA                     DS   CL(MQCFIL_LENGTH)
```

Visual Basic language declaration

```
Type MQCFIL
  Type As Long          ' Structure type
  StrucLength As Long   ' Structure length
  Parameter As Long     ' Parameter identifier
  Count As Long         ' Count of parameter value
End Type
```

MQCFIL64 - 64-bit integer list parameter

Use this page to view the structure of an MQCFIL64 parameter and the declarations for the following programming languages: C, COBOL, PL/I, RPG/ILE, and S/390 assembler

The MQCFIL64 structure describes a 64-bit integer list parameter. Following the links to the declarations is a description of the fields making up the MQCFIL64 structure:

- C language
- COBOL language
- PL/I language (z/OS only)
- RPG/ILE language (IBM i only)
- System/390 assembler-language (z/OS only)

Type

Description: Indicates that the structure is a MQCFIL64 structure describing a 64-bit integer list parameter.

Data type: MQLONG.

Value:

MQCFT_INTEGER64_LIST
Structure defining a 64-bit integer list.

StrucLength

Description: Length in bytes of the MQCFIL64 structure, including the array of integers at the end of the structure (the *Values* field).

Data type: MQLONG.

Parameter

Description: Identifies the parameter with a value that is contained in the structure.

Data type: MQLONG.

Count

Description: Number of elements in the *Values* array.

Data type: MQLONG.

Values: 0 or greater.

Values

Description: Array of values for the parameter identified by the *Parameter* field.

Data type: (MQINT64 x *Count*)

The way that this field is declared depends on the programming language:

- For the C programming language, the field is declared as an array with one element. Storage for the structure must be allocated dynamically, and pointers used to address the fields within it.
- For the COBOL, PL/I, RPG, and System/390 assembler programming languages, the field is omitted from the structure declaration. When an instance of the structure is declared, you must include MQCFIL64 in a larger structure, and declare additional fields following MQCFIL64, to represent the *Values* field as required.

For COBOL, additional fields should be declared as:

```
    PIC  S9(18)
```

For PL/I, additional fields should be declared as FIXED BINARY SIGNED with a precision of 63.

For System/390 assembler, additional fields should be declared D (double word) in the DS declaration.

C language declaration

```
typedef struct tagMQCFIN64 {
  MQLONG  Type;         /* Structure type */
  MQLONG  StrucLength;  /* Structure length */
  MQLONG  Parameter;    /* Parameter identifier */
  MQLONG  Count;        /* Count of parameter values */
  MQINT64 Values[1];        /* Parameter value */
 } MQCFIL64;
```

COBOL language declaration

```
**   MQCFIL64 structure
  10 MQCFIL64.
**    Structure type
   15 MQCFIL64-TYPE        PIC S9(9) BINARY.
**    Structure length
   15 MQCFIL64-STRUCLENGTH PIC S9(9) BINARY.
**    Parameter identifier
   15 MQCFIL64-PARAMETER   PIC S9(9) BINARY.
**    Count of parameter values
   15 MQCFIL64-COUNT       PIC S9(9) BINARY.
```

PL/I language declaration

```
dcl
 1 MQCFIL64 based,
  3 Type        fixed bin(31), /* Structure type */
  3 StrucLength fixed bin(31), /* Structure length */
  3 Parameter   fixed bin(31), /* Parameter identifier */
  3 Count       fixed bin(31)  /* Count of parameter values */
```

RPG/ILE language declaration (IBM i only)

```
     D*..1....:....2....:....3....:....4....:....5....:....6....:....7..
     D* MQCFIL64 Structure
     D*
     D* Structure type
     D  IL64TYP               1      4I 0 INZ(25)
     D* Structure length
     D  IL64LEN               5      8I 0 INZ(16)
     D* Parameter identifier
     D  IL64PRM               9     12I 0 INZ(0)
     D* Count of parameter values
```

```
     D  IL64CNT               13     16I 0 INZ(0)
     D* Parameter values -- first element
     D  IL64VAL               17     16    INZ(0)
```

S/390 assembler-language declaration (z/OS only)

```
MQCFIL64                        DSECT
MQCFIL64_TYPE                   DS   F        Structure type
MQCFIL64_STRUCLENGTH            DS   F        Structure length
MQCFIL64_PARAMETER              DS   F        Parameter identifier
MQCFIL64_COUNT                  DS   F        Parameter value high
MQCFIL64_LENGTH                 EQU  *-MQCFIL64 Length of structure
                                ORG  MQCFIL64
MQCFIL64_AREA                   DS   CL(MQCFIL64_LENGTH)
```

MQCFIN - Integer parameter

Use this page to view the structure of an MQCFIN parameter and the declarations for the following programming languages: C, COBOL, PL/I, RPG/ILE, S/390 assembler, and Visual Basic

The MQCFIN structure describes an integer parameter. Following the links to the declarations is a description of the fields making up the MQCFIN structure:

- C language
- COBOL language
- PL/I language (z/OS only)
- RPG/ILE language (IBM i only)
- S/390 assembler-language (z/OS only)
- Visual Basic language (Windows only)

Type

Description: Indicates that the structure type is MQCFIN and describes an integer parameter.

Data type: MQLONG.

Value:

MQCFT_INTEGER
Structure defining an integer.

StrucLength

Description: Length in bytes of the MQCFIN structure.

Data type: MQLONG.

Value:

MQCFIN_STRUC_LENGTH
Length of MQCFIN structure.

Parameter

Description: Identifies the parameter with a value that is contained in the structure.

Data type: MQLONG.

Value

Description: Value of parameter identified by the *Parameter* field.

Data type: MQLONG.

C language declaration

```
typedef struct tagMQCFIN {
  MQLONG  Type;         /* Structure type */
  MQLONG  StrucLength;  /* Structure length */
  MQLONG  Parameter;    /* Parameter identifier */
  MQLONG  Value;        /* Parameter value */
 } MQCFIN;
```

COBOL language declaration

```
**   MQCFIN structure
  10 MQCFIN.
**    Structure type
   15 MQCFIN-TYPE        PIC S9(9) BINARY.
**    Structure length
   15 MQCFIN-STRUCLENGTH PIC S9(9) BINARY.
**    Parameter identifier
   15 MQCFIN-PARAMETER   PIC S9(9) BINARY.
**    Parameter value
   15 MQCFIN-VALUE       PIC S9(9) BINARY.
```

PL/I language declaration

```
dcl
 1 MQCFIN based,
  3 Type        fixed bin(31), /* Structure type */
  3 StrucLength fixed bin(31), /* Structure length */
  3 Parameter   fixed bin(31), /* Parameter identifier */
  3 Value       fixed bin(31); /* Parameter value */
```

RPG/ILE declaration (IBM i only)

```
     D*..1....:....2....:....3....:....4....:....5....:....6....:....7..
     D* MQCFIN Structure
     D*
     D* Structure type
     D  INTYP                  1      4I 0
     D* Structure length
     D  INLEN                  5      8I 0
     D* Parameter identifier
     D  INPRM                  9     12I 0
     D* Parameter value
     D  INVAL                 13     16I 0
```

S/390 assembler-language declaration

```
MQCFIN                         DSECT
MQCFIN_TYPE                    DS   F        Structure type
MQCFIN_STRUCLENGTH             DS   F        Structure length
MQCFIN_PARAMETER               DS   F        Parameter identifier
MQCFIN_VALUE                   DS   F        Parameter value
MQCFIN_LENGTH                  EQU  *-MQCFIN Length of structure
                               ORG  MQCFIN
MQCFIN_AREA                    DS   CL(MQCFIN_LENGTH)
```

Visual Basic language declaration

```
Type MQCFIN
  Type As Long         ' Structure type
  StrucLength As Long  ' Structure length
  Parameter As Long    ' Parameter identifier
  Value As Long        ' Parameter value
End Type
```

MQCFIN64 - 64-bit integer parameter

Use this page to view the structure of an MQCFIN64 parameter and the declarations for the following programming languages: C, COBOL, PL/I, RPG/ILE, and S/390 assembler

The MQCFIN64 structure describes a 64-bit integer parameter. Following the links to the declarations is a description of the fields making up the MQCFIN64 structure:

- C language
- COBOL language
- PL/I language (z/OS only)
- RPG/ILE language (IBM i only)
- System/390 assembler-language (z/OS only)

Type

Description: Indicates that the structure is a MQCFIN64 structure describing a 64-bit integer parameter.

Data type: MQLONG.

Value:

MQCFT_INTEGER64
Structure defining a 64-bit integer.

StrucLength

Description: Length in bytes of the MQCFIN64 structure.

Data type: MQLONG.

Value:

MQCFIN64_STRUC_LENGTH
Length of 64-bit integer parameter structure.

Parameter

Description: Identifies the parameter with a value that is contained in the structure.

Data type: MQLONG.

Values

Description: This is the value of the parameter identified by the *Parameter* field.

Data type: (MQINT64)

C language declaration

```
typedef struct tagMQCFIN64 {
  MQLONG  Type;         /* Structure type */
  MQLONG  StrucLength;  /* Structure length */
  MQLONG  Parameter;    /* Parameter identifier */
  MQLONG  Reserved;     /* Reserved */
  MQINT64 Value;        /* Parameter value */
 } MQCFIN64;
```

COBOL language declaration

```
**   MQCFIN64 structure
  10 MQCFIN64.
**    Structure type
   15 MQCFIN64-TYPE        PIC S9(9) BINARY.
**    Structure length
   15 MQCFIN64-STRUCLENGTH PIC S9(9) BINARY.
**    Parameter identifier
```

```
   15 MQCFIN64-PARAMETER   PIC S9(9) BINARY.
**    Reserved
   15 MQCFIN64-RESERVED    PIC S9(9) BINARY.
**    Parameter value
   15 MQCFIN64-VALUE       PIC S9(18) BINARY.
```

PL/I language declaration

```
dcl
 1 MQCFIN64 based,
  3 Type        fixed bin(31), /* Structure type */
  3 StrucLength fixed bin(31), /* Structure length */
  3 Parameter   fixed bin(31), /* Parameter identifier */
  3 Reserved    fixed bin(31)  /* Reserved */
  3 Value       fixed bin(63); /* Parameter value */
```

RPG/ILE language declaration (IBM i only)

```
     D*..1....:....2....:....3....:....4....:....5....:....6....:....7..
     D* MQCFIN64 Structure
     D*
     D* Structure type
     D  IN64TYP                1      4I 0 INZ(23)
     D* Structure length
     D  IN64LEN                5      8I 0 INZ(24)
     D* Parameter identifier
     D  IN64PRM                9     12I 0 INZ(0)
     D* Reserved field
     D  IN64RSV               13     16I 0 INZ(0)
     D* Parameter value
     D  IN64VAL               17     16    INZ(0)
```

S/390 assembler-language declaration (z/OS only)

```
MQCFIN64                        DSECT
MQCFIN64_TYPE                   DS   F        Structure type
MQCFIN64_STRUCLENGTH            DS   F        Structure length
MQCFIN64_PARAMETER              DS   F        Parameter identifier
MQCFIN64_RESERVED               DS   F        Reserved
MQCFIN64_VALUE                  DS   D        Parameter value
MQCFIN64_LENGTH                 EQU  *-MQCFIN64 Length of structure
                                ORG  MQCFIN64
MQCFIN64_AREA                   DS   CL(MQCFIN64_LENGTH)
```

MQCFSL - String list parameter

Use this page to view the structure of an MQCFSL parameter and the declarations for the following programming languages: COBOL, PL/I, RPG/ILE, S/390 assembler, and Visual Basic

The MQCFSL structure describes a string list parameter. Following the links to the declarations is a description of the fields making up the MQCFSL structure:

- COBOL language
- PL/I language (z/OS only)
- RPG/ILE language (IBM i only)
- System/390 assembler-language (z/OS only)
- Visual Basic language (Windows only)

Type

Description: This indicates that the structure is an MQCFSL structure describing a string-list parameter.

Data type: MQLONG.

Value:

MQCFT_STRING_LIST
Structure defining a string list.

StrucLength

Description: This is the length in bytes of the MQCFSL structure, including the array of strings at the end of the structure (the *Strings* field).

Data type: MQLONG.

Parameter

Description: This identifies the parameter with values that are contained in the structure.

Data type: MQLONG.

CodedCharSetId

Description: This specifies the coded character set identifier of the data in the *Strings* field.

Data type: MQLONG.

Count

Description: This is the number of strings present in the *Strings* field; zero or greater.

Data type: MQLONG.

StringLength

Description: This is the length in bytes of one parameter value, that is the length of one string in the *Strings* field; all of the strings are this length.

Data type: MQLONG.

String

Description: This is a set of string values for the parameter identified by the *Parameter* field. The number of strings is given by the *Count* field, and the length of each string is given by the *StringLength* field. The strings are concatenated together, with no bytes skipped between adjacent strings. The total length of the strings is the length of one string multiplied by the number of strings present (that is, *StringLength* x *Count*).

In MQFMT_EVENT messages, trailing blanks can be omitted from string parameters (that is, the string may be shorter than the defined length of the parameter). *StringLength* gives the length of the string actually present in the message.

Note: In the MQCFSL structure, a null character in a string is treated as normal data, and does not act as a delimiter for the string. This means that when a receiving application reads a MQFMT_EVENT message, the receiving application receives all of the data specified by the sending application. The data may, of course, have been converted between character sets (for example, by the receiving application specifying the MQGMO_CONVERT option on the MQGET call).

Data type: MQCHAR x *StringLength* x *Count*

COBOL language declaration

```
**   MQCFSL structure
  10 MQCFSL.
**    Structure type
   15 MQCFSL-TYPE           PIC S9(9) BINARY.
**    Structure length
   15 MQCFSL-STRUCLENGTH    PIC S9(9) BINARY.
**    Parameter identifier
   15 MQCFSL-PARAMETER      PIC S9(9) BINARY.
**    Coded character set identifier
   15 MQCFSL-CODEDCHARSETID PIC S9(9) BINARY.
**    Count of parameter values
   15 MQCFSL-COUNT          PIC S9(9) BINARY.
**    Length of one string
   15 MQCFSL-STRINGLENGTH   PIC S9(9) BINARY.
```

PL/I language declaration

```
dcl
 1 MQCFSL based,
  3 Type           fixed bin(31), /* Structure type */
  3 StrucLength    fixed bin(31), /* Structure length */
  3 Parameter      fixed bin(31), /* Parameter identifier */
  3 CodedCharSetId fixed bin(31), /* Coded character set identifier */
  3 Count          fixed bin(31), /* Count of parameter values */
  3 StringLength   fixed bin(31); /* Length of one string */
```

RPG/ILE declaration (IBM i only)

```
     D*..1....:....2....:....3....:....4....:....5....:....6....:....7..
     D* MQCFSL Structure
     D*
     D* Structure type
     D  SLTYP                  1      4I 0
     D* Structure length
     D  SLLEN                  5      8I 0
     D* Parameter identifier
     D  SLPRM                  9     12I 0
     D* Coded character set identifier
     D  SLCSI                 13     16I 0
     D* Count of parameter values
     D  SLCNT                 17     20I 0
     D* Length of one string
     D  SLSTL                 21     24I 0
```

S/390 assembler-language declaration (z/OS only)

```
MQCFSL                 DSECT
MQCFSL_TYPE            DS   F  Structure type
MQCFSL_STRUCLENGTH     DS   F  Structure length
MQCFSL_PARAMETER       DS   F  Parameter identifier
MQCFSL_CODEDCHARSETID  DS   F  Coded character set identifier
MQCFSL_COUNT           DS   F  Count of parameter values
MQCFSL_STRINGLENGTH    DS   F  Length of one string
*
MQCFSL_LENGTH          EQU  *-MQCFSL
                       ORG  MQCFSL
MQCFSL_AREA            DS   CL(MQCFSL_LENGTH)
```

Visual Basic language declaration (Windows systems only)

```
Type MQCFSL
  Type           As Long 'Structure type'
  StrucLength    As Long 'Structure length'
  Parameter      As Long 'Parameter identifier'
  CodedCharSetId As Long 'Coded character set identifier'
  Count          As Long 'Count of parameter values'
  StringLength   As Long 'Length of one string'
End Type
```

MQCFST - String parameter

Use this page to view the structure of an MQCFST parameter and the declarations for the following programming languages: C, COBOL, PL/I, RPG/ILE, S/390 assembler, and Visual Basic

The MQCFST structure describes a string parameter. Following the links to the declarations is a description of the fields making up the MQCFST structure:

- C language
- COBOL language
- PL/I language (z/OS only)
- RPG/ILE language (IBM i only)
- System/390 assembler-language (z/OS only)
- Visual Basic language (Windows only)

The MQCFST structure ends with a variable-length character string; see the *String* field for further details.

Type

Description: Indicates that the structure type is MQCFST and describes a string parameter.

Data type: MQLONG.

Value:

MQCFT_STRING
Structure defining a string.

StrucLength

Description: Length in bytes of the MQCFST structure, including the string at the end of the structure (the *String* field).

Data type: MQLONG.

Parameter

Description: Identifies the parameter with a value that is contained in the structure.

Data type: MQLONG.

Values: Dependent on the event message.

CodedCharSetId

Description: Coded character set identifier of the data in the *String* field.

Data type: MQLONG.

StringLength

Description: Length in bytes of the data in the *String* field; zero or greater.

Data type: MQLONG.

String

Description: The value of the parameter identified by the *Parameter* field.

In MQFMT_EVENT messages, trailing blanks can be omitted from string parameters (that is, the string may be shorter than the defined length of the parameter). *StringLength* gives the length of the string actually present in the message.

Data type: MQCHAR x *StringLength*

Value: The string can contain any characters that are in the character set defined by *CodedCharSetId*, and that are valid for the parameter identified by *Parameter*.

Language considerations: The way that this field is declared depends on the programming language:

- For the C programming language, the field is declared as an array with one element. Storage for the structure should be allocated dynamically, and pointers used to address the fields within it.
- For the COBOL, PL/I, System/390 assembler, and Visual Basic programming languages, the field is omitted from the structure declaration. When an instance of the structure is declared, the user should include MQCFST in a larger structure, and declare additional fields following MQCFST, to represent the *String* field as required.

A null character in the string is treated as normal data, and does not act as a delimiter for the string. This means that when a receiving application reads an MQFMT_EVENT message, the receiving application receives all of the data specified by the sending application. The data may, of course, have been converted between character sets (for example, by the receiving application specifying the MQGMO_CONVERT option on the MQGET call).

C language declaration

```
typedef struct tagMQCFST {
  MQLONG  Type;            /* Structure type */
  MQLONG  StrucLength;     /* Structure length */
  MQLONG  Parameter;       /* Parameter identifier */
  MQLONG  CodedCharSetId;  /* Coded character set identifier */
  MQLONG  StringLength;    /* Length of string */
  MQCHAR  String[1];       /* String value - first
                              character */
 } MQCFST;
```

COBOL language declaration

```
**   MQCFST structure
  10 MQCFST.
**    Structure type
   15 MQCFST-TYPE           PIC S9(9) BINARY.
**    Structure length
   15 MQCFST-STRUCLENGTH    PIC S9(9) BINARY.
**    Parameter identifier
   15 MQCFST-PARAMETER      PIC S9(9) BINARY.
**    Coded character set identifier
   15 MQCFST-CODEDCHARSETID PIC S9(9) BINARY.
**    Length of string
   15 MQCFST-STRINGLENGTH   PIC S9(9) BINARY.
```

PL/I language declaration

```
dcl
 1 MQCFST based,
  3 Type           fixed bin(31), /* Structure type */
  3 StrucLength    fixed bin(31), /* Structure length */
  3 Parameter      fixed bin(31), /* Parameter identifier */
  3 CodedCharSetId fixed bin(31), /* Coded character set identifier */
  3 StringLength   fixed bin(31); /* Length of string */
```

RPG/ILE declaration (IBM i only)

```
     D*..1....:....2....:....3....:....4....:....5....:....6....:....7..
     D* MQCFST Structure
```

```
     D*
     D* Structure type
     D  STTYP                  1      4I 0
     D* Structure length
     D  STLEN                  5      8I 0
     D* Parameter identifier
     D  STPRM                  9     12I 0
     D* Coded character set identifier
     D  STCSI                 13     16I 0
     D* Length of string
     D  STSTL                 17     20I 0
```

S/390 assembler-language declaration

```
MQCFST                          DSECT
MQCFST_TYPE                     DS   F        Structure type
MQCFST_STRUCLENGTH              DS   F        Structure length
MQCFST_PARAMETER                DS   F        Parameter identifier
MQCFST_CODEDCHARSETID           DS   F        Coded character set
*                                             identifier
MQCFST_STRINGLENGTH             DS   F        Length of string
MQCFST_LENGTH                   EQU  *-MQCFST Length of structure
                                ORG  MQCFST
MQCFST_AREA                     DS   CL(MQCFST_LENGTH)
```

Visual Basic language declaration

```
Type MQCFST
  Type As Long            ' Structure type
  StrucLength As Long     ' Structure length
  Parameter As Long       ' Parameter identifier
  CodedCharSetId As Long  ' Coded character set identifier
  StringLength As Long    ' Length of string
End Type
```

MQEPH - Embedded PCF header

Use this page to view the structure of an MQEPH embedded PCF header and the declarations for the following programming languages: C, COBOL, PL/I, RPG/ILE, S/390 assembler, and Visual Basic

The MQEPH structure describes the additional data that is present in a message when that message is a programmable command format (PCF) message. Following the links to the declarations is a description of the fields making up the MQEPH structure:

- C language
- COBOL language
- PL/I language (z/OS only)
- RPG/ILE language (IBM i only)
- S/390 assembler-language (z/OS only)
- Visual Basic language (Windows only)

The additional data consists of the MQEPH structure followed by an array of PCF parameter structures. To include the MQEPH structure in a message, the **Format** parameter in the message descriptor is set to MQFMT_EMBEDDED.

StrucId

Description: Structure identifier.

Data type: MQCHAR4.

Value:

MQEPH_STRUC_ID
Identifier for distribution header structure.

Version

Description: Structure version number.

Data type: MQLONG.

Value:

MQEPH_VERSION_1
Version number for embedded PCF header structure.

StrucLength

Description: Structure length. This is the length in bytes of the MQEPH structure and is set to the amount of data preceding the next header structure.

Data type: MQLONG.

Encoding

Description: Numeric encoding. This specifies the numeric encoding of the data that follows the last PCF parameter structure.

Data type: MQLONG.

CodedCharSetId

Description: Coded character set identifier. This specifies the coded character set identifier of the data that follows the last PCF parameter structure.

Data type: MQLONG.

Format

Description: Format. This specifies the format name of the data that follows the last PCF parameter structure.

Data type: MQCHAR8.

Flags

Description: Flags. This is a reserved field.

Data type: MQLONG.

Value:

MQEPH_NONE
No flags have been specified.

MQEPH_CCSID_EMBEDDED
The character set of the parameters containing character data is specified individually within the CodedCharSetId field in each structure. The character set of the StrucId and Format fields is defined by the CodedCharSetId field in the header structure that precedes the MQEPH structure, or by the CodedCharSetId field in the MQMD if the MQEPH is at the start of the message.

PCFHeader

Description: Command format header.

Data type: MQCFH.

C language declaration

```
struct tagMQEPH  {
  MQCHAR4 StrucId;         /* Structure identifier */
  MQLONG  Version;         /* Structure version number */
  MQLONG  StrucLength      /* Structure length */
  MQLONG  Encoding;        /* Numeric encoding */
  MQLONG  CodedCharSetId;  /* Coded character set identifier */
  MQCHAR8 Format;          /* Data format */
  MQLONG  Flags;           /* Flags */
  MQCFH   PCFHeader;       /* PCF header */
 } MQEPH;
```

COBOL language declaration

```
**   MQEPH structure
  10 MQEPH.
**    Structure identifier
   15 MQEPH-STRUCID        PIC X(4).
**    Structure version number
   15 MQEPH-VERSION        PIC S9(9) BINARY.
**    Structure length
   15 MQEPH-STRUCLENGTH    PIC S9(9) BINARY.
**    Numeric encoding
   15 MQEPH-ENCODING       PIC S9(9) BINARY.
**    Coded characeter set identifier
   15 MQEPH-CODEDCHARSETID PIC S9(9) BINARY.
**    Data format
   15 MQEPH-FORMAT         PIC X(8).
**    Flags
   15 MQEPH-FLAGS          PIC S9(9) BINARY.
**    PCF header
   15 MQEPH-PCFHEADER.
**    Structure type
    20 MQEPH-PCFHEADER-TYPE            PIC S9(9) BINARY.
**    Structure length
    20 MQEPH-PCFHEADER-STRUCLENGTH     PIC S9(9) BINARY.
**    Structure version number
    20 MQEPH-PCFHEADER-VERSION         PIC S9(9) BINARY.
**    Command identifier
    20 MQEPH-PCFHEADER-COMMAND         PIC S9(9) BINARY.
**    Message sequence number
    20 MQEPH-PCFHEADER-MSGSEQNUMBER    PIC S9(9) BINARY.
**    Control options
    20 MQEPH-PCFHEADER-CONTROL         PIC S9(9) BINARY.
**    Completion code
    20 MQEPH-PCFHEADER-COMPCODE        PIC S9(9) BINARY.
**    Reason code qualifying completion code
    20 MQEPH-PCFHEADER-REASON          PIC S9(9) BINARY.
**    Count of parameter structures
    20 MQEPH-PCFHEADER-PARAMETERCOUNT  PIC S9(9) BINARY.
```

PL/I language declaration (z/OS and Windows)

```
dcl
 1 MQEPH based,
  3 StrucId         char(4),       /* Structure identifier */
  3 Version         fixed bin(31), /* Structure version number */
  3 StrucLength     fixed bin(31), /* Structure length */
  3 Encoding        fixed bin(31), /* Numeric encoding */
  3 CodedCharSetId  fixed bin(31), /* Coded character set identifier */
  3 Format          char(8),       /* Data format */
  3 Flags           fixed bin(31), /* Flags */
  3 PCFHeader,                     /* PCF header */
   5 Type           fixed bin(31), /* Structure type */
   5 StrucLength    fixed bin(31), /* Structure length */
   5 Version        fixed bin(31), /* Structure version number */
   5 Command        fixed bin(31), /* Command identifier */
   5 MsgSeqNumber   fixed bin(31), /* Message sequence number */
   5 Control        fixed bin(31), /* Control options */
   5 CompCode       fixed bin(31), /* Completion code */
   5 Reason         fixed bin(31), /* Reason code qualifying completion
                                      code */
   5 ParameterCount fixed bin(31); /* Count of parameter structures */
```

RPG language declaration (IBM i only)

```
D*..1....:....2....:....3....:....4....:....5....:....6....:....7..
D* MQEPH Structure
D*
D* Structure identifier
D  EPSID                  1      4    INZ('EPH ')
D* Structure version number
D  EPVER                  5      8I 0 INZ(1)
D* Structure length
D  EPLEN                  9     12I 0 INZ(68)
D* Numeric encoding
D  EPENC                 13     16I 0 INZ(0)
D* Coded character set identifier
D  EPCSI                 17     20I 0 INZ(0)
D* Format name
D  EPFMT                 21     28I 0 INZ('        ')
D* Flags
D  EPFLG                 29     32I 0 INZ(0)
D* Programmable Command Format Header
D*
D* Structure type
D  EP1TYPE               33     36I 0 INZ(0)
D* Structure length
D  EP1LEN                37     40I 0 INZ(36)
D* Structure version number
D  EP1VER                41     44I 0 INZ(3)
D* Command identifier
D  EP1CMD                45     48I 0 INZ(0)
D* Message sequence number
D  EP1SEQ                49     52I 0 INZ(1)
D* Control options
D  EP1CTL                53     56I 0 INZ(1)
D* Completion code
D  EP1CMP                57     60I 0 INZ(0)
D* Reason code qualifying completion code
D  EP1REA                61     64I 0 INZ(0)
D* Count of parameter structures
D  EP1CNT                65     68I 0 INZ(0)
```

S/390 assembler-language declaration (z/OS only)

```
MQEPH                          DSECT
MQEPH_STRUCID                  DS   CL4      Structure identifier
MQEPH_VERSION                  DS   F        Structure version number
MQEPH_STRUCLENGTH              DS   F        Structure length
MQEPH_ENCODING                 DS   F        Numeric encoding
MQEPH_CODEDCHARSETID           DS   F        Coded character set identifier
MQEPH_FORMAT                   DS   CL8      Data format
MQEPH_FLAGS                    DS   F        Flags
MQEPH_PCFHEADER                DS   0F       Force fullword alignment
MQEPH_PCFHEADER_TYPE           DS   F        Structure type
MQEPH_PCFHEADER_STRUCLENGTH    DS   F        Structure length
MQEPH_PCFHEADER_VERSION        DS   F        Structure version number
MQEPH_PCFHEADER_COMMAND        DS   F        Command identifier
MQEPH_PCFHEADER_MSGSEQNUMBER   DS   F        Message sequence number
MQEPH_PCFHEADER_CONTROL        DS   F        Control options
MQEPH_PCFHEADER_COMPCODE       DS   F        Completion code
MQEPH_PCFHEADER_REASON         DS   F        Reason code qualifying completion code
MQEPH_PCFHEADER_PARAMETERCOUNT DS   F        Count of parameter structures
MQEPH_PCFHEADER_LENGTH         EQU  *-MQEPH_PCFHEADER
                               ORG  MQEPH_PCFHEADER
MQEPH_PCFHEADER_AREA           DS   CL(MQEPH_PCFHEADER_LENGTH)
*
MQEPH_LENGTH                   EQU  *-MQEPH
                               ORG  MQEPH
MQEPH_AREA                     DS   CL(MQEPH_LENGTH)
```

Visual Basic language declaration (Windows only)

```
Type MQEPH
  StrucId As String*4     'Structure identifier
  Version As Long         'Structure version number
  StrucLength As Long     'Structure length
  Encoding As Long        'Numeric encoding
  CodedCharSetId As Long  'Coded characetr set identifier
```

```
   Format As String*8      'Format name
   Flags As Long           'Flags
   Reason As Long          'Reason code qualifying completion code
   PCFHeader As MQCFH      'PCF header
End Type
```

Object attributes for event data

Information about the object attributes that IBM MQ monitoring techniques can include in the configuration event data recorded in event messages. The amount of event data depends on the type of object to which the configuration event relates.

Authentication information attributes

Event messages relating to objects can include authentication information attributes

AlterationDate (MQCFST)
Alteration date (parameter identifier: **MQCA_ALTERATION_DATE**).

The date when the information was last altered.

AlterationTime (MQCFST)
Alteration time (parameter identifier: **MQCA_ALTERATION_TIME**).

The time when the information was last altered.

AuthInfoConnName (MQCFST)
Authentication information connection name (parameter identifier: **MQCA_AUTH_INFO_CONN_NAME**).

The maximum length of the string is 48.

AuthInfoDesc (MQCFST)
Authentication information description (parameter identifier: **MQCA_AUTH_INFO_DESC**).

The maximum length of the string is MQ_AUTH_INFO_DESC_LENGTH.

AuthInfoName (MQCFST)
Authentication information object name (parameter identifier: **MQCA_AUTH_INFO_NAME**).

The maximum length of the string is MQ_AUTH_INFO_NAME_LENGTH.

AuthInfoType (MQCFIN)
Authentication information type (parameter identifier: **MQIA_AUTH_INFO_TYPE**).

The value is MQAIT_CRL_LDAP.

LDAPPassword (MQCFST)
LDAP password (parameter identifier: **MQCA_LDAP_PASSWORD**).

The maximum length of the string is MQ_LDAP_PASSWORD_LENGTH.

LDAPUserName (MQCFST)
LDAP user name (parameter identifier: **MQCA_LDAP_USER_NAME**).

The maximum length of the string is 256.

CF structure attributes

Event messages relating to objects can include CF structure attributes

AlterationDate (MQCFST)
Alteration date (parameter identifier: **MQCA_ALTERATION_DATE**).

The date when the information was last altered.

AlterationTime (MQCFST)
Alteration time (parameter identifier: **MQCA_ALTERATION_TIME**).

The time when the information was last altered.

CFLevel (MQCFIN)
CF level (parameter identifier: **MQIA_CF_LEVEL**).

CFStrucDesc (MQCFST)
CF Structure description (parameter identifier: **MQCA_CF_STRUC_DESC**).

The maximum length of the string is MQCA_CF_STRUC_DESC_LENGTH.

Recovery (MQCFIN)
Recovery (parameter identifier: **MQIA_CF_RECOVER**).

Communication information attributes

AlterationDate (MQCFST)
Alteration date (parameter identifier: **MQCA_ALTERATION_DATE**).

The date when the information was last altered, in the form *yyyy-mm-dd*.

AlterationTime (MQCFST)
Alteration time (parameter identifier: **MQCA_ALTERATION_TIME**).

The time when the information was last altered, in the form *hh.mm.ss*.

Bridge (MQCFIN)
Bridge (parameter identifier: **MQIA_MCAST_BRIDGE**).

Specifies whether publications from applications not using Multicast are bridged to applications using multicast.

The value can be any of the following values:

MQMCB_DISABLED
Bridging is disabled.

MQMCB_ENABLED
Bridging is enabled.

CCSID (MQCFIN)
Coded character set identifier (parameter identifier: **MQIA_CODED_CHAR_SET_ID**).

The CCSID that messages are transmitted on.

CommEvent (MQCFIN)
Communication event (parameter identifier: **MQIA_COMM_EVENT**).

Controls whether event messages are generated for multicast handles that are created using this COMMINFO object.

The value can be any of the following values:

MQEVR_DISABLED
Event messages are not generated.

MQEVR_ENABLED
Event messages are generated.

MQEVR_EXCEPTION
Event messages are generated if the message reliability is below the reliability threshold.

CominfoName (MQCFST)
Communication information name (parameter identifier: **MQCA_COMM_INFO_NAME**).

The name of the administrative communication information definition about which information is to be returned.

Description (MQCFST)
Description (parameter identifier: **MQCA_COMM_INFO_DESC**).

Plain-text comment that provides descriptive information about the communication information object.

Encoding (MQCFIN)
Encoding (parameter identifier: **MQIACF_ENCODING**).

The encoding that the messages are transmitted in.

The value can be any of the following values:

MQENC_AS_PUBLISHED

MQENC_NORMAL

MQENC_REVERSED

MQENC_S390

MQENC_TNS

GrpAddress (MQCFST)
Group address (parameter identifier: **MQCACH_GROUP_ADDRESS**).

The group IP address or DNS name.

MonitorInterval (MQCFIN)
Frequency of monitoring (parameter identifier: **MQIA_MONITOR_INTERVAL**).

How frequently, in seconds, monitoring information is updated and event messages are generated.

MulticastHeartbeat (MQCFIN)
Multicast heartbeat (parameter identifier: **MQIACH_MC_HB_INTERVAL**).

Heartbeat interval measured in milliseconds.

MulticastPropControl (MQCFIN)
Multicast properties control (parameter identifier: **MQIACH_MULTICAST_PROPERTIES**).

Controls how many of the MQMD properties and user properties flow with the message.

The value can be any of the following values:

MQMCP_ALL
All properties are transmitted.

MQMCP_REPLY
Only user properties and MQMD fields that deal with replying to the messages are transmitted.

MQMCP_USER
Only user properties are transmitted.

MQMCP_NONE
No properties are transmitted.

MQMCP_COMPAT
Properties are transmitted in a format compatible with previous IBM MQ multicast clients.

MsgHistory (MQCFIN)
Message history (parameter identifier: **MQIACH_MSG_HISTORY**).

The amount of message history in kilobytes that is kept by the system to handle retransmissions in the case of NACKs.

NewSubHistory (MQCFIN)
New Subscriber History (parameter identifier: **MQIACH_NEW_SUBSCRIBER_HISTORY**).

Controls how much historical data a new subscriber receives. The value can be any of the following values:

MQNSH_NONE
Only publications from the time of the subscription are sent.

MQNSH_ALL
As much history as is known is retransmitted.

PortNumber (MQCFIN)
Port Number (parameter identifier: **MQIACH_PORT**).

The port number to transmit on.

Type (MQCFIN)
Type (parameter identifier: **MQIA_COMM_INFO_TYPE**).

The type of the communications information object.

Channel attributes

Event messages relating to objects can include channel attributes

Only those attributes that apply to the type of channel in question are included in the event data.

AlterationDate (MQCFST)
Alteration date (parameter identifier: **MQCA_ALTERATION_DATE**).

The date when the information was last altered.

AlterationTime (MQCFST)
Alteration time (parameter identifier: **MQCA_ALTERATION_TIME**).

The time when the information was last altered.

BatchHeartbeat (MQCFIN)
The value being used for the batch heartbeating (parameter identifier: **MQIACH_BATCH_HB**).

The value can be in the range 0 through 999999. A value of 0 indicates heartbeating is not in use.

BatchInterval (MQCFIN)
Batch interval (parameter identifier: **MQIACH_BATCH_INTERVAL**).

BatchSize (MQCFIN)
Batch size (parameter identifier: **MQIACH_BATCH_SIZE**).

***ChannelDesc* (MQCFST)**
Channel description (parameter identifier: MQCACH_DESC).

The maximum length of the string is MQ_CHANNEL_DESC_LENGTH.

ChannelMonitoring (MQCFIN)
Level of monitoring data collection for the channel (parameter identifier: **MQIA_MONITORING_CHANNEL**).

The value can be any of the following values:

MQMON_OFF
Monitoring data collection is turned off.

MQMON_LOW
Monitoring data collection is turned on with a low ratio of data collection.

MQMON_MEDIUM
Monitoring data collection is turned on with a medium ratio of data collection.

MQMON_HIGH
Monitoring data collection is turned on with a high ratio of data collection.

MQMON_Q_MGR
The level of monitoring data collected is based on the queue manager attribute **ChannelMonitoring**.

ChannelName (MQCFST)
Channel name (parameter identifier: **MQCACH_CHANNEL_NAME**).

The maximum length of the string is MQ_CHANNEL_NAME_LENGTH.

ChannelStatistics (MQCFIN)
Level of statistics data collection for the channel (parameter identifier: **MQIA_STATISTICS_CHANNEL**).

The value can be any of the following values:

MQMON_OFF
Statistics data collection is turned off.

MQMON_LOW
Statistics data collection is turned on with a low ratio of data collection.

MQMON_MEDIUM
Statistics data collection is turned on with a medium ratio of data collection.

MQMON_HIGH
Statistics data collection is turned on with a high ratio of data collection.

MQMON_Q_MGR
The level of statistics data collected is based on the queue manager attribute **ChannelStatistics**.

For platform specific details about this attribute, refer to the description of the STATCHL attribute in the ALTER QMGR command.

ChannelType (MQCFIN)
Channel type (parameter identifier: **MQIACH_CHANNEL_TYPE**).

The value can be:

MQCHT_SENDER
Sender.

MQCHT_SERVER
Server.

MQCHT_RECEIVER
Receiver.

MQCHT_REQUESTER
Requester.

MQCHT_SVRCONN
Server-connection (for use by clients).

MQCHT_CLNTCONN
Client connection.

MQCHT_CLUSRCVR
Cluster-receiver.

MQCHT_CLUSSDR
Cluster-sender.

CipherSpec (MQCFST)
SSL cipher specification (parameter identifier: **MQCACH_SSL_CIPHER_SPEC**).

The maximum length of the string is MQ_SSL_CIPHER_SPEC_LENGTH.

ClusterName (MQCFST)
Cluster name (parameter identifier: **MQCA_CLUSTER_NAME**).

ClusterNamelist (MQCFST)
Cluster namelist (parameter identifier: **MQCA_CLUSTER_NAMELIST**).

CLWLChannelPriority (MQCFIN)
Cluster workload channel priority (parameter identifier: **MQIACH_CLWL_CHANNEL_PRIORITY**).

CLWLChannelRank (MQCFIN)
Cluster workload channel rank (parameter identifier: **MQIACH_CLWL_CHANNEL_RANK**).

CLWLChannelWeight (MQCFIN)
Cluster workload channel weight (parameter identifier: **MQIACH_CLWL_CHANNEL_WEIGHT**).

ConnectionName (MQCFST)
Connection name (parameter identifier: **MQCACH_CONNECTION_NAME**).

The maximum length of the string is MQ_CONN_NAME_LENGTH.

DataConversion (MQCFIN)
Whether sender should convert application data (parameter identifier: **MQIACH_DATA_CONVERSION**).

The value can be any of the following values:

MQCDC_NO_SENDER_CONVERSION
No conversion by sender.

MQCDC_SENDER_CONVERSION
Conversion by sender.

DiscInterval (MQCFIN)
Disconnection interval (parameter identifier: **MQIACH_DISC_INTERVAL**).

HeaderCompression (MQCFIL)
Header data compression techniques supported by the channel (parameter identifier: **MQIACH_HDR_COMPRESSION**).

For sender, server, cluster-sender, cluster-receiver, and client-connection channels, the values specified are in order of preference.

The value can be one, or more, of the following:

MQCOMPRESS_NONE
No header data compression is performed.

MQCOMPRESS_SYSTEM
Header data compression is performed.

HeartbeatInterval (MQCFIN)
Heartbeat interval (parameter identifier: **MQIACH_HB_INTERVAL**).

KeepAliveInterval (MQCFIN)
Keep alive interval (parameter identifier: **MQIACH_KEEP_ALIVE_INTERVAL**).

LocalAddress (MQCFST)
Local communications address for the channel (parameter identifier: **MQCACH_LOCAL_ADDRESS**).

The maximum length of the string is MQ_LOCAL_ADDRESS_LENGTH.

LongRetryCount (MQCFIN)
Long retry count (parameter identifier: **MQIACH_LONG_RETRY**).

LongRetryInterval (MQCFIN)
Long timer (parameter identifier: **MQIACH_LONG_TIMER**).

MaxMsgLength (MQCFIN)
Maximum message length (parameter identifier: **MQIACH_MAX_MSG_LENGTH**).

MCAName (MQCFST)
Message channel agent name (parameter identifier: **MQCACH_MCA_NAME**).

The maximum length of the string is MQ_MCA_NAME_LENGTH.

MCAType (MQCFIN)
Message channel agent type (parameter identifier: **MQIACH_MCA_TYPE**).

The value can be any of the following values:

MQMCAT_PROCESS
Process

MQMCAT_THREAD
Thread

MCAUserIdentifier (MQCFST)
Message channel agent user identifier (parameter identifier: **MQCACH_MCA_USER_ID**).

The maximum length of the MCA user identifier is MQ_MCA_USER_ID_LENGTH.

MessageCompression (MQCFIL)
Message data compression techniques supported by the channel (parameter identifier: **MQIACH_MSG_COMPRESSION**).

For sender, server, cluster-sender, cluster-receiver, and client-connection channels, the values specified are in order of preference.

The value can be one, or more, of:

MQCOMPRESS_NONE
No message data compression is performed. This is the default value.

MQCOMPRESS_RLE
Message data compression is performed using run-length encoding.

MQCOMPRESS_ZLIBFAST
Message data compression is performed using ZLIB encoding with speed prioritized.

MQCOMPRESS_ZLIBHIGH
Message data compression is performed using ZLIB encoding with compression prioritized.

MQCOMPRESS_ANY
Any compression technique supported by the queue manager can be used. This is only valid for receiver, requester, and server-connection channels.

ModeName (MQCFST)
Mode name (parameter identifier: **MQCACH_MODE_NAME**).

The maximum length of the string is MQ_MODE_NAME_LENGTH.

MsgExit (MQCFSL)
Message exit name (parameter identifier: **MQCACH_MSG_EXIT_NAME**).

The number of names in the list is given by the **Count** field in the MQCFSL structure. It is the same as the count for **MsgUserData**. It may exceed the number of exit names specified for the channel, in which case the excess names are blank; the minimum is 1. The length of each name is given by the **StringLength** field in that structure.

The maximum length of the exit name is MQ_EXIT_NAME_LENGTH.

MsgRetryCount (MQCFIN)
Message retry count (parameter identifier: **MQIACH_MR_COUNT**).

Specifies the number of times that a failing message should be retried.

This parameter is only valid for receiver, cluster-receiver, and requester channels.

MsgRetryExit (MQCFST)
Message retry exit name (parameter identifier: **MQCACH_MR_EXIT_NAME**).

This parameter is only valid for receiver, cluster-receiver, and requester channels.

The maximum length of the string is MQ_MAX_EXIT_NAME_LENGTH.

MsgRetryInterval (MQCFIN)
Message retry interval (parameter identifier: **MQIACH_MR_INTERVAL**).

Specifies the minimum time interval in milliseconds between retries of failing messages.

This parameter is only valid for receiver, cluster-receiver, and requester channels.

MsgRetryUserData (MQCFST)
Message retry exit user data (parameter identifier: **MQCACH_MR_EXIT_USER_DATA**).

Specifies user data that is passed to the message retry exit.

This parameter is only valid for receiver, cluster-receiver, and requester channels.

The maximum length of the string is MQ_EXIT_DATA_LENGTH.

MsgUserData (MQCFSL)
Message exit user data (parameter identifier: **MQCACH_MSG_EXIT_USER_DATA**).

The number of names in the list is given by the **Count** field in the MQCFSL structure. It is the same as the count for **MsgExit**. The length of each name is given by the **StringLength** field in that structure.

The maximum length of the string is MQ_EXIT_DATA_LENGTH.

NetworkPriority (MQCFIN)
Network priority (parameter identifier: **MQIACH_NETWORK_PRIORITY**).

NonPersistentMsgSpeed (MQCFIN)
Speed at which nonpersistent messages are to be sent (parameter identifier: **MQIACH_NPM_SPEED**).

The value can be any of the following values:

MQNPMS_NORMAL
Normal speed.

MQNPMS_FAST
Fast speed.

Password (MQCFST)
Password (parameter identifier: **MQCACH_PASSWORD**).

The maximum length of the string is MQ_PASSWORD_LENGTH.

PeerName (MQCFST)
SSL peer name (parameter identifier: **MQCACH_SSL_PEER_NAME**).

The maximum length of the string is 256.

PutAuthority (MQCFIN)
Put authority (parameter identifier: **MQIACH_PUT_AUTHORITY**).

The value can be:

MQPA_DEFAULT
Default user identifier is used.

MQPA_CONTEXT
Context user identifier is used.

MQPA_ALTERNATE_OR_MCA
Alternate or MCA user identifier is used.

MQPA_ONLY_MCA
Only MCA user identifier is used.

QMgrName (MQCFST)
Queue manager name (parameter identifier: **MQCA_Q_MGR_NAME**).

The maximum length of the string is MQ_Q_MGR_NAME_LENGTH.

ReceiveExit (MQCFSL)
Receive exit name (parameter identifier: **MQCACH_RCV_EXIT_NAME**).

The number of names in the list is given by the **Count** field in the MQCFSL structure. It is the same as the count for **ReceiveUserData**. It may exceed the number of exit names specified for the channel, in which case the excess names are blank; the minimum is 1. The length of each name is given by the **StringLength** field in that structure.

For a client-connection channel the maximum length of the exit name is MQ_MAX_EXIT_NAME_LENGTH. For all other channels, the maximum length of the exit name is MQ_EXIT_NAME_LENGTH.

ReceiveUserData (MQCFSL)
Receive exit user data (parameter identifier: **MQCACH_RCV_EXIT_USER_DATA**).

The number of names in the list is given by the **Count** field in the MQCFSL structure. It is the same as the count for **ReceiveExit**. The length of each name is given by the **StringLength** field in that structure.

The maximum length of the string is MQ_EXIT_DATA_LENGTH.

SecurityExit (MQCFST)
Security exit name (parameter identifier: **MQCACH_SEC_EXIT_NAME**).

For a client-connection channel the maximum length of the exit name is MQ_MAX_EXIT_NAME_LENGTH. For all other channels, the maximum length of the exit name is MQ_EXIT_NAME_LENGTH.

SecurityUserData (MQCFST)
Security exit user data (parameter identifier: **MQCACH_SEC_EXIT_USER_DATA**).

The maximum length of the string is MQ_EXIT_DATA_LENGTH.

SendExit (MQCFSL)
Send exit name (parameter identifier: **MQCACH_SEND_EXIT_NAME**).

The number of names in the list is given by the **Count** field in the MQCFSL structure. It is the same as the count for **SendUserData**. It may exceed the number of exit names specified for the channel, in which case the excess names are blank; the minimum is 1. The length of each name is given by the *StringLength* field in that structure.

For a client-connection channel the maximum length of the exit name is MQ_MAX_EXIT_NAME_LENGTH. For all other channels, the maximum length of the exit name is MQ_EXIT_NAME_LENGTH.

SendUserData (MQCFSL)
Send exit user data (parameter identifier: **MQCACH_SEND_EXIT_USER_DATA**).

The number of names in the list is given by the **Count** field in the MQCFSL structure. It is the same as the count for **SendExit**. The length of each name is given by the **StringLength** field in that structure.

The maximum length of the string is MQ_EXIT_DATA_LENGTH.

SeqNumberWrap (MQCFIN)
Sequence wrap number (parameter identifier: **MQIACH_SEQUENCE_NUMBER_WRAP**).

ShortRetryCount (MQCFIN)
Short retry count (parameter identifier: **MQIACH_SHORT_RETRY**).

ShortRetryInterval (MQCFIN)
Short timer (parameter identifier: **MQIACH_SHORT_TIMER**).

SSLClientAuthentication (MQCFIN)
SSL client authentication (parameter identifier: **MQIACH_SSL_CLIENT_AUTH**).

The value can be:

MQSCA_REQUIRED
Certificate required.

MQSCA_OPTIONAL
Certificate optional.

TpName (MQCFST)
Transaction program name (parameter identifier: **MQCACH_TP_NAME**).

The maximum length of the string is MQ_TP_NAME_LENGTH.

TransportType (MQCFIN)
Transmission protocol type (parameter identifier: **MQIACH_XMIT_PROTOCOL_TYPE**).

The value may be:

MQXPT_LU62
LU 6.2.

MQXPT_TCP
TCP.

MQXPT_NETBIOS
NetBIOS.

MQXPT_SPX
SPX.

UserIdentifier (MQCFST)
Task user identifier (parameter identifier: **MQCACH_USER_ID**).

The maximum length of the string is MQ_USER_ID_LENGTH.

XmitQName (MQCFST)
Transmission queue name (parameter identifier: **MQCACH_XMIT_Q_NAME**).

The maximum length of the string is MQ_Q_NAME_LENGTH.

Channel authentication attributes

Event messages relating to objects can include channel authentication attributes

Only those attributes that apply to the type of channel in question are included in the event data.

ChannelProfile (MQCFST)
Channel Profile (parameter identifier: **MQCACH_CHANNEL_NAME**).

Maximum length is MQ_CHANNEL_NAME_LENGTH.

Returned: Always.

ChannelAuthType (MQCFIN)
Channel Authentication Type (parameter identifier: **MQIACF_CHLAUTH_TYPE**).

Returned: Always.

Warning (MQCFIN)
Warning (parameter identifier: **MQIACH_WARNING**).

Returned: Always.

connectionNameList (MQCFSL)
Connection Name List (parameter identifier: **MQCACH_CONNECTION_NAME_LIST**).

Element length: MQ_CONN_NAME_LENGTH.

Returned: Only when **ChannelAuthType** is MQAUT_BLOCKADDR.

MCAUserIdList (MQCFSL)
MCA User Id List (parameter identifier: **MQCACH_MCA_USER_ID_LIST**).

Element length: MQ_MCA_USER_ID_LENGTH.

Returned: Only when **ChannelAuthType** is MQAUT_BLOCKUSER.

MCAUser (MQCFST)
MCA User (parameter identifier: **MQCACH_MCA_USER_ID**).

Maximum length: MQ_MCA_USER_ID_LENGTH.

Returned: Only when **ChannelAuthType** is of a mapping type (MQCAUT_SSLPEERMAP, MQCAUT_ADDRESSMAP, MQCAUT_USERMAP or MQCAUT_QMGRMAP).

ConnectionName (MQCFST)
Connection Name (parameter identifier: **MQCACH_CONNECTION_NAME**).

Maximum length: MQ_CONN_NAME_LENGTH.

Returned: Only when **ChannelAuthType** is of a mapping type (MQCAUT_SSLPEERMAP, MQCAUT_ADDRESSMAP, MQCAUT_USERMAP or MQCAUT_QMGRMAP).

UserSource (MQCFIN)
User Source (parameter identifier: **MQIACH_USER_SOURCE**).

Returned: Only when **ChannelAuthType** is of a mapping type (MQCAUT_SSLPEERMAP, MQCAUT_ADDRESSMAP, MQCAUT_USERMAP or MQCAUT_QMGRMAP).

SSLPeerName (MQCFST)
SSL Peer Name (parameter identifier: **MQCACH_SSL_PEER_NAME**).

Maximum length: MQ_SSL_PEER_NAME_LENGTH.

Returned: Only when **ChannelAuthType** is MQCAUT_SSLPEERMAP.

ClientUserId (MQCFST)
Client User Id (parameter identifier: **MQCACH_CLIENT_USER_ID**).

Maximum length: MQ_MCA_USER_ID_LENGTH.

Returned: Only when **ChannelAuthType** is MQCAUT_USERMAP.

RemoteQueueManagerName (MQCFST)
Remote Queue Manager Name (parameter identifier: **MQCA_REMOTE_Q_MGR_NAME**).

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: Only when **ChannelAuthType** is MQCAUT_QMGRMAP.

Listener attributes

AlterationDate (MQCFST)
Alteration date (parameter identifier: **MQCA_ALTERATION_DATE**).

The date, in the form *yyyy-mm-dd*, on which the information was last altered.

AlterationTime (MQCFST)
Alteration time (parameter identifier: **MQCA_ALTERATION_TIME**).

The time, in the form *hh.mm.ss*, at which the information was last altered.

Windows **Adapter (MQCIN)**
Adapter number (parameter identifier: **MQIACH_ADAPTER**).

The adapter number on which NetBIOS listens. This parameter is valid only on Windows.

Backlog (MQCIN)
Backlog (parameter identifier: **MQIACH_BACKLOG**).

The number of concurrent connection requests that the listener supports.

Windows **Commands (MQCIN)**
Adapter number (parameter identifier: **MQIACH_COMMAND_COUNT**).

The number of commands that the listener can use. This parameter is valid only on Windows.

IPAddress (MQCFST)
IP address (parameter identifier: **MQCACH_IP_ADDRESS**).

IP address for the listener specified in IPv4 dotted decimal, IPv6 hexadecimal notation, or alphanumeric host name form.

ListenerDesc (MQCFST)
Description of listener definition (parameter identifier: **MQCACH_LISTENER_DESC**).

ListenerName (MQCFST)
Name of listener definition (parameter identifier: **MQCACH_LISTENER_NAME**).

Windows **LocalName (MQCFST)**
NetBIOS local name (parameter identifier: **MQCACH_LOCAL_NAME**).

The NetBIOS local name that the listener uses. This parameter is valid only on Windows.

Windows **NetbiosNames (MQCFIN)**
NetBIOS names (parameter identifier: **MQIACH_NAME_COUNT**).

The number of names that the listener supports. This parameter is valid only on Windows.

Port (MQCFIN)
Port number (parameter identifier: **MQIACH_PORT**).

The port number for TCP/IP. This parameter is valid only if the value of **TransportType** is MQXPT_TCP.

Windows **Sessions (MQCFIN)**
NetBIOS sessions (parameter identifier: MQIACH_SESSION_COUNT).

The number of sessions that the listener can use. This parameter is valid only on Windows.

Socket (MQCFIN)
SPX socket number (parameter identifier: **MQIACH_SOCKET**).

The SPX socket on which to listen. This parameter is valid only if the value of **TransportType** is MQXPT_SPX.

StartMode (MQCFIN)
Service mode (parameter identifier: **MQIACH_LISTENER_CONTROL**).

Specifies how the listener is to be started and stopped. The value can be:

MQSVC_CONTROL_MANUAL
The listener is started and stopped manually, by user command.

MQSVC_CONTROL_Q_MGR
The listener is started and stopped when the queue manager starts and stops.

MQSVC_CONTROL_Q_MGR_START
The listener is started when the queue manager starts, but does not stop when the queue manager stops.

Windows **TPName (MQCFST)**
Transaction program name (parameter identifier: **MQCACH_TP_NAME**).

The LU 6.2 transaction program name. This parameter is valid only on Windows.

TransportType (MQCFIN)
Transmission protocol (parameter identifier: **MQIACH_XMIT_PROTOCOL_TYPE**).

The value can be any of the following values:

MQXPT_TCP
TCP

MQXPT_LU62
LU 6.2

MQXPT_NETBIOS
NetBIOS

MQXPT_SPX
SPX

Namelist attributes

Event messages relating to objects can include namelist attributes

AlterationDate (MQCFST)
Alteration date (parameter identifier: **MQCA_ALTERATION_DATE**).

The date when the information was last altered.

AlterationTime (MQCFST)
Alteration time (parameter identifier: **MQCA_ALTERATION_TIME**).

The time when the information was last altered.

NameCount (MQCFIN)
Number of names in the namelist (parameter identifier: **MQIA_NAME_COUNT**).

The number of names contained in the namelist.

NamelistDesc (MQCFST)
Description of namelist definition (parameter identifier: **MQCA_NAMELIST_DESC**).

The maximum length of the string is MQ_NAMELIST_DESC_LENGTH.

NamelistName (MQCFST)
The name of the namelist definition (parameter identifier: **MQCA_NAMELIST_NAME**).

The maximum length of the string is MQ_NAMELIST_NAME_LENGTH.

NamelistType (MQCFIN)
Namelist type (parameter identifier: **MQIA_NAMELIST_TYPE**).

Names (MQCFSL)
The names contained in the namelist (parameter identifier: **MQCA_NAMES**).

The number of names in the list is given by the **Count** field in the MQCFSL structure. The length of each name is given by the **StringLength** field in that structure. The maximum length of a name is MQ_OBJECT_NAME_LENGTH.

Process attributes

Event messages relating to objects can include process attributes

AlterationDate (MQCFST)
Alteration date (parameter identifier: **MQCA_ALTERATION_DATE**).

The date when the information was last altered.

AlterationTime (MQCFST)
Alteration time (parameter identifier: **MQCA_ALTERATION_TIME**).

The time when the information was last altered.

ApplId (MQCFST)
Application identifier (parameter identifier: **MQCA_APPL_ID**).

The maximum length of the string is MQ_PROCESS_APPL_ID_LENGTH.

ApplType (MQCFIN)
Application type (parameter identifier: **MQIA_APPL_TYPE**).

EnvData (MQCFST)
Environment data (parameter identifier: **MQCA_ENV_DATA**).

The maximum length of the string is MQ_PROCESS_ENV_DATA_LENGTH.

ProcessDesc (MQCFST)
Description of process definition (parameter identifier: **MQCA_PROCESS_DESC**).

The maximum length of the string is MQ_PROCESS_DESC_LENGTH.

ProcessName (MQCFST)
The name of the process definition (parameter identifier: **MQCA_PROCESS_NAME**).

The maximum length of the string is MQ_PROCESS_NAME_LENGTH.

UserData (MQCFST)
User data (parameter identifier: **MQCA_USER_DATA**).

The maximum length of the string is MQ_PROCESS_USER_DATA_LENGTH.

Queue attributes

Event messages relating to objects can include queue attributes

Only those attributes that apply to the type of queue in question are included in the event data.

AlterationDate (MQCFST)
Alteration date (parameter identifier: **MQCA_ALTERATION_DATE**).

The date when the information was last altered.

AlterationTime (MQCFST)
Alteration time (parameter identifier: **MQCA_ALTERATION_TIME**).

The time when the information was last altered.

BackoutRequeueName (MQCFST)
Excessive backout requeue name (parameter identifier: **MQCA_BACKOUT_REQ_Q_NAME**).

The maximum length of the string is MQ_Q_NAME_LENGTH.

BackoutThreshold (MQCFIN)
Backout threshold (parameter identifier: **MQIA_BACKOUT_THRESHOLD**).

BaseQName (MQCFST)
Queue name to which the alias resolves (parameter identifier: **MQCA_BASE_Q_NAME**).

This is the name of a queue that is defined to the local queue manager.

The maximum length of the string is MQ_Q_NAME_LENGTH.

CFstructure (MQCFST)
CF structure name (parameter identifier: **MQCA_CF_STRUC_NAME**).

The maximum length of the string is MQ_CF_STRUC_NAME_LENGTH.

ClusterName (MQCFST)
Cluster name (parameter identifier: **MQCA_CLUSTER_NAME**).

ClusterNamelist (MQCFST)
Cluster namelist (parameter identifier: **MQCA_CLUSTER_NAMELIST**).

CLWLQueuePriority (MQCFIN)
Queue priority (parameter identifier: **MQIA_CLWL_Q_PRIORITY**).

CLWLQueueRank (MQCFIN)
Queue rank (parameter identifier: **MQIA_CLWL_Q_RANK**).

CLWLUseQ (MQCFIN)
This defines the behavior of an MQPUT when the target queue has both a local instance and at least one remote cluster instance (parameter identifier: **MQIA_CLWL_USEQ**).

The value can be any of the following values:

MQCLWL_USEQ_ANY
Use remote and local queues.

MQCLWL_USEQ_LOCAL
Do not use remote queues.

MQCLWL_USEQ_AS_Q_MGR
Inherit definition from the queue manager attribute **CLWLUseQ**.

CreationDate (MQCFST)
Queue creation date (parameter identifier: **MQCA_CREATION_DATE**).

The maximum length of the string is MQ_CREATION_DATE_LENGTH.

CreationTime (MQCFST)
Creation time (parameter identifier: **MQCA_CREATION_TIME**).

The maximum length of the string is MQ_CREATION_TIME_LENGTH.

DefBind (MQCFIN)
Default binding (parameter identifier: **MQIA_DEF_BIND**).

The value can be:

MQBND_BIND_ON_OPEN
Binding fixed by MQOPEN call.

MQBND_BIND_NOT_FIXED
Binding not fixed.

MQBND_BIND_ON_GROUP
Allows an application to request that a group of messages are all allocated to the same destination instance.

DefinitionType (MQCFIN)
Queue definition type (parameter identifier: **MQIA_DEFINITION_TYPE**).

The value can be any of the following values:

MQQDT_PREDEFINED
Predefined permanent queue.

MQQDT_PERMANENT_DYNAMIC
Dynamically defined permanent queue.

MQQDT_SHARED_DYNAMIC
Dynamically defined permanent queue that is shared.

DefInputOpenOption (MQCFIN)
Default input open option for defining whether queues can be shared (parameter identifier: **MQIA_DEF_INPUT_OPEN_OPTION**).

The value can be:

MQOO_INPUT_EXCLUSIVE
Open queue to get messages with exclusive access.

MQOO_INPUT_SHARED
Open queue to get messages with shared access.

DefPersistence (MQCFIN)
Default persistence (parameter identifier: **MQIA_DEF_PERSISTENCE**).

The value can be any of the following values:

MQPER_PERSISTENT
Message is persistent.

MQPER_NOT_PERSISTENT
Message is not persistent.

DefPriority (MQCFIN)
Default priority (parameter identifier: **MQIA_DEF_PRIORITY**).

HardenGetBackout (MQCFIN)
Whether to harden backout (parameter identifier: **MQIA_HARDEN_GET_BACKOUT**).

The value can be any of the following values:

MQQA_BACKOUT_HARDENED
Backout count remembered.

MQQA_BACKOUT_NOT_HARDENED
Backout count may not be remembered.

IndexType (MQCFIN)
Index type (parameter identifier: **MQIA_INDEX_TYPE**).

InhibitGet (MQCFIN)
Whether get operations are allowed (parameter identifier: **MQIA_INHIBIT_GET**).

The value can be any of the following values:

MQQA_GET_ALLOWED
Get operations are allowed.

MQQA_GET_INHIBITED
Get operations are inhibited.

InhibitPut (MQCFIN)
Whether put operations are allowed (parameter identifier: **MQIA_INHIBIT_PUT**).

The value can be any of the following values:

MQQA_PUT_ALLOWED
Put operations are allowed.

MQQA_PUT_INHIBITED
Put operations are inhibited.

InitiationQName (MQCFST)
Initiation queue name (parameter identifier: **MQCA_INITIATION_Q_NAME**).

The maximum length of the string is MQ_Q_NAME_LENGTH.

MaxMsgLength (MQCFIN)
Maximum message length (parameter identifier: **MQIA_MAX_MSG_LENGTH**).

MaxQDepth (MQCFIN)
Maximum queue depth (parameter identifier: **MQIA_MAX_Q_DEPTH**).

MsgDeliverySequence (MQCFIN)
Whether priority is relevant (parameter identifier: **MQIA_MSG_DELIVERY_SEQUENCE**).

The value can be any of the following values:

MQMDS_PRIORITY
Messages are returned in priority order.

MQMDS_FIFO
Messages are returned in FIFO order (first in, first out).

ProcessName (MQCFST)
Name of process definition for queue (parameter identifier: **MQCA_PROCESS_NAME**).

The maximum length of the string is MQ_PROCESS_NAME_LENGTH.

QDepthHiEvent (MQCFIN)
Controls whether Queue Depth High events are generated. (parameter identifier: **MQIA_Q_DEPTH_HIGH_EVENT**).

The value can be any of the following values:

MQEVR_ENABLED
Queue depth high events are enabled.

MQEVR_DISABLED
Queue depth high events are disabled.

QDepthHighLimit (MQCFIN)
High limit for queue depth (parameter identifier: **MQIA_Q_DEPTH_HIGH_LIMIT**).

The threshold against which the queue depth is compared to generate a Queue Depth High event.

QDepthLoEvent (MQCFIN)
Controls whether Queue Depth Low events are generated. (parameter identifier: **MQIA_Q_DEPTH_LOW_EVENT**).

The value can be any of the following values:

MQEVR_ENABLED
Queue depth lowevents are enabled.

MQEVR_DISABLED
Queue depth low events are disabled.

QDepthLowLimit (MQCFIN)
Low limit for queue depth (parameter identifier: **MQIA_Q_DEPTH_LOW_LIMIT**).

The threshold against which the queue depth is compared to generate a Queue Depth Low event.

QDepthMaxEvent (MQCFIN)
Controls whether Queue Full events are generated. (parameter identifier: **MQIA_Q_DEPTH_MAX_EVENT**).

The value can be any of the following values:

MQEVR_ENABLED
Queue depth full events are enabled.

MQEVR_DISABLED
Queue depth full events are disabled.

QDesc (MQCFST)
Queue description (parameter identifier: **MQCA_Q_DESC**).

The maximum length of the string is MQ_Q_DESC_LENGTH.

QName (MQCFST)
Queue name (parameter identifier: **MQCA_Q_NAME**).

The maximum length of the string is MQ_Q_NAME_LENGTH.

QServiceInterval (MQCFIN)
Target for queue service interval (parameter identifier: **MQIA_Q_SERVICE_INTERVAL**).

The service interval used for comparison to generate Queue Service Interval High and Queue Service Interval OK events.

QType (MQCFIN)
Queue type (parameter identifier: **MQIA_Q_TYPE**).

The value can be:

MQQT_ALIAS
Alias queue definition.

MQQT_LOCAL
Local queue.

MQQT_REMOTE
Local definition of a remote queue.

MQQT_MODEL
Model queue definition.

QueueAccounting (MQCFIN)
Specifies whether accounting information is collected (parameter identifier: **MQIA_ACCOUNTING_Q**).

The value can be any of the following values:

MQMON_ON
Accounting information is collected for the queue.

MQMON_OFF
Accounting information is not collected for the queue.

MQMON_Q_MGR
The collection of accounting information for this queue is based is based on the queue manager attribute **QueueAccounting**.

QueueMonitoring (MQCFIN)
Level of monitoring data collection for the queue (parameter identifier: **MQIA_MONITORING_Q**).

The value can be any of the following values:

MQMON_OFF
Monitoring data collection is turned off.

MQMON_LOW
Monitoring data collection is turned on with a low ratio of data collection.

MQMON_MEDIUM
Monitoring data collection is turned on with a moderate ratio of data collection.

MQMON_HIGH
Monitoring data collection is turned on with a high ratio of data collection.

MQMON_Q_MGR
The level of monitoring data collected is based on the queue manager attribute **QueueMonitoring**.

RemoteQMgrName (MQCFST)
Name of remote queue manager (parameter identifier: **MQCA_REMOTE_Q_MGR_NAME**).

The maximum length of the string is MQ_Q_MGR_NAME_LENGTH.

RemoteQName (MQCFST)
Name of remote queue as known locally on the remote queue manager (parameter identifier: **MQCA_REMOTE_Q_NAME**).

The maximum length of the string is MQ_Q_NAME_LENGTH.

RetentionInterval (MQCFIN)
Retention interval (parameter identifier: **MQIA_RETENTION_INTERVAL**).

ServiceIntervalEvent (MQCFIN)
Controls whether Service Interval High or Service Interval OK events are generated. .

The value can be any of the following values:

MQQSIE_NONE
No service interval events are generated.

MQQSIE_OK
Service interval OK events are generated.

MQQSIE_HIGH
Service interval high events are generated.

Shareability (MQCFIN)
Whether queue can be shared (parameter identifier: **MQIA_SHAREABILITY**).

The value can be any of the following values:

MQQA_SHAREABLE
Queue is shareable.

MQQA_NOT_SHAREABLE
Queue is not shareable.

StorageClass (MQCFST)
Storage class name (parameter identifier: **MQCA_STORAGE_CLASS**).

The maximum length of the string is MQ_STORAGE_CLASS_LENGTH.

TriggerControl (MQCFIN)
Trigger control (parameter identifier: **MQIA_TRIGGER_CONTROL**).

The value can be any of the following values:

MQTC_OFF
Trigger messages not required.

MQTC_ON
Trigger messages required.

TriggerData (MQCFST)
Trigger data (parameter identifier: **MQCA_TRIGGER_DATA**).

The maximum length of the string is MQ_TRIGGER_DATA_LENGTH.

TriggerDepth (MQCFIN)
Trigger depth (parameter identifier: **MQIA_TRIGGER_DEPTH**).

TriggerMsgPriority (MQCFIN)
Threshold message priority for triggers (parameter identifier: **MQIA_TRIGGER_MSG_PRIORITY**).

TriggerType (MQCFIN)
Trigger type (parameter identifier: **MQIA_TRIGGER_TYPE**).

The value can be:

MQTT_NONE
No trigger messages.

MQTT_FIRST
Trigger message when queue depth goes from 0 to 1.

MQTT_EVERY
Trigger message for every message.

MQTT_DEPTH
Trigger message when depth threshold exceeded.

Usage (MQCFIN)
Usage (parameter identifier: **MQIA_USAGE**).

The value can be any of the following values:

MQUS_NORMAL
Normal usage.

MQUS_TRANSMISSION
Transmission queue.

XmitQName (MQCFST)
Transmission queue name (parameter identifier: **MQCA_XMIT_Q_NAME**).

The maximum length of the string is MQ_Q_NAME_LENGTH.

Queue manager attributes

Event messages relating to objects can include queue manager attributes.

AccountingConnOverride (MQCFIN)
Specifies whether applications can override the settings of the **QueueAccounting** and **MQIAccounting** queue manager parameters (parameter identifier: **MQIA_ACCOUNTING_CONN_OVERRIDE**).

The value can be any of the following values:

MQMON_DISABLED
Applications cannot override the settings of the **QueueAccounting** and **MQIAccounting** parameters.

This value is the initial default value for the queue manager.

MQMON_ENABLED
Applications can override the settings of the **QueueAccounting** and **MQIAccounting** parameters by using the options field of the MQCNO structure of the MQCONNX API call.

This parameter is valid only on IBM i, AIX, Linux, and Windows.

Multi **AccountingInterval (MQCFIN)**
The time interval, in seconds, at which intermediate accounting records are written (parameter identifier: **MQIA_ACCOUNTING_INTERVAL**).

Specify a value in the range 1 - 604,000.

This parameter is valid only on IBM i, AIX, Linux, and Windows.

ActivityRecording (MQCFIN)
Specifies whether activity recording is enabled or disabled (parameter identifier: **MQIA_ACTIVITY_RECORDING**).

The value can be any of the following values:

MQRECORDING_MSG
Activity recording is enabled. Activity reports are delivered to the reply-to queue specified in the message descriptor of the message.

MQRECORDING_Q
Activity recording is enabled. Activity reports are delivered to a fixed name queue.

MQRECORDING_DISABLED
Activity recording is disabled.

AdoptNewMCACheck (MQCFIN)
Procedure to determine if an existing receiver MCA is to be adopted when an inbound channel is detected of the same name (parameter identifier: **MQIA_ADOPTNEWMCA_CHECK**).

The value can be any of the following values:

MQADOPT_CHECK_Q_MGR_NAME
Compare the receiver MCA and the inbound channel. If the queue manager names match, the existing receiver MCA is adopted providing it is active. If they don't match, the existing receiver MCA is canceled, and a new MCA is created.

MQADOPT_CHECK_NET_ADDR
Compare the receiver MCA and the inbound channel. If the network addresses match, the existing receiver MCA is adopted providing it is active. If they don't match, the existing receiver MCA is canceled, and a new MCA is created.

MQADOPT_CHECK_ALL
Compare the receiver MCA and the inbound channel. If both the queue manager names, and the network addresses match, the existing receiver MCA is adopted providing it is active. If they don't match, the existing receiver MCA is canceled, and a new MCA is created.

MQADOPT_CHECK_NONE
If the existing receiver MCA is active it is adopted with no checks.

AdoptNewMCAType (MQCFIN)
Specifies whether orphaned receiver MCAs are to be restarted when an inbound channel matching the **AdoptNewMCACheck** procedure is detected (parameter identifier: **MQIA_ADOPTNEWMCA_TYPE**).

The value can be:

MQADOPT_TYPE_NO
Do not restart and adopt orphaned receiver MCAs.

MQADOPT_TYPE_ALL
Restart and adopt orphaned receiver MCAs.

AlterationDate (MQCFST)
Alteration date (parameter identifier: **MQCA_ALTERATION_DATE**).

The date when the information was last altered.

AlterationTime (MQCFST)
Alteration time (parameter identifier: **MQCA_ALTERATION_TIME**).

The time when the information was last altered.

AuthorityEvent (MQCFIN)
Controls whether authorization (Not Authorized) events are generated (parameter identifier: **MQIA_AUTHORITY_EVENT**).

The value can be any of the following values:

MQEVR_ENABLED
Authorization event reporting enabled.

MQEVR_DISABLED
Authorization event reporting disabled.

BridgeEvent (MQCFIN)
Determines whether IMS bridge events are generated (parameter identifier: **MQIA_BRIDGE_EVENT**).

The value can be any of the following values:

MQEVR_ENABLED
All IMS bridge events are enabled.

MQEVR_DISABLED
All IMS bridge events are disabled.

CertificateLabel (MQCFST)
Specifies the certificate label for this queue manager to use. The label identifies which personal certificate in the key repository has been selected (parameter identifier: **MQCA_CERT_LABEL**).

ALW **CertificateValPolicy (MQCFIN)**
Specifies which TLS certificate validation policy is used to validate digital certificates received from remote partner systems (parameter identifier: **MQIA_CERT_VAL_POLICY**).

This attribute can be used to control how strictly the certificate chain validation conforms to industry security standards. For more information, see Certificate validation policies in IBM MQ.

The value can be any of the following values:

MQ_CERT_VAL_POLICY_ANY
Apply each of the certificate validation policies supported by the secure sockets library and accept the certificate chain if any of the policies considers the certificate chain valid. This setting can be used for maximum backwards compatibility with older digital certificates which do not comply with the modern certificate standards.

MQ_CERT_VAL_POLICY_RFC5280
Apply only the RFC 5280 compliant certificate validation policy. This setting provides stricter validation than the ANY setting, but rejects some older digital certificates.

This parameter is only valid on AIX, Linux, and Windows and can be used only on a queue manager with a command level of 711, or higher.

Changes to **CertificateValPolicy** become effective in the following cases:

- When a new channel process is started.
- For channels that run as threads of the channel initiator, when the channel initiator is restarted.
- For channels that run as threads of the listener, when the listener is restarted.
- For channels that run as threads of a process pooling process, when the process pooling process is started or restarted and first runs a TLS channel. If the process pooling process has already run a TLS channel, and you want the change to become effective immediately, run the MQSC command **REFRESH SECURITY TYPE(SSL)**. The process pooling process is **amqrmppa** on AIX, Linux, and Windows.
- When a **REFRESH SECURITY TYPE(SSL)** command is issued.

z/OS **CFConlos (MQCFIN)**
Specifies the action to be taken when the queue manager loses connectivity to the administration structure, or any CF structure with **CFConlos** set to ASQMGR (parameter identifier: **MQIA_QMGR_CFCONLOS**).

The value can be:

MQCFCONLOS_TERMINATE
The queue manager terminates when connectivity to CF structures is lost.

MQCFCONLOS_TOLERATE
The queue manager tolerates loss of connectivity to CF structures without terminating.

This parameter applies to z/OS only.

ChannelAuthenticationRecords (MQCFIN)
Controls whether channel authentication records are used (parameter identifier: **MQIA_CHLAUTH_RECORDS**).

Channel authentication records can be set and displayed regardless of the value of this attribute.

The value can be any of the following values:

MQCHLA_DISABLED
Channel authentication records are not checked.

MQCHLA_ENABLED
Channel authentication records are checked.

Multi **ChannelAutoDef (MQCFIN)**
Controls whether receiver and server-connection channels can be auto-defined (parameter identifier: **MQIA_CHANNEL_AUTO_DEF**).

Auto-definition for cluster-sender channels is always enabled.

This parameter is valid only on IBM i, AIX, Linux, and Windows.

The value can be:

MQCHAD_DISABLED
Channel auto-definition disabled.

MQCHAD_ENABLED
Channel auto-definition enabled.

Multi **ChannelAutoDefEvent (MQCFIN)**
Controls whether channel auto-definition events are generated (parameter identifier: **MQIA_CHANNEL_AUTO_DEF_EVENT**), when a receiver, server-connection, or cluster-sender channel is auto-defined.

This parameter is valid only on IBM i, AIX, Linux, and Windows.

The value can be:

MQEVR_DISABLED
Event reporting disabled.

MQEVR_ENABLED
Event reporting enabled.

ChannelAutoDefExit (MQCFST)
Channel auto-definition exit name (parameter identifier: **MQCA_CHANNEL_AUTO_DEF_EXIT**).

The maximum length of the exit name is **MQ_EXIT_NAME_LENGTH**.

This parameter is supported only in the environments in which an MQSeries® 5.1 or later product is available.

ChannelEvent (MQCFIN)
Determines whether channel events are generated (parameter identifier: **MQIA_CHANNEL_EVENT**).

The value can be any of the following values:

MQEVR_ENABLED
All channel events are enabled.

MQEVR_EXCEPTION
Only the following channels events are enabled:

- MQRC_CHANNEL_ACTIVATED
- MQRC_CHANNEL_CONV_ERROR
- MQRC_CHANNEL_NOT_ACTIVATED
- MQRC_CHANNEL_STOPPED

MQEVR_DISABLED
All channel events are disabled.

Multi **ChannelInitiatorControl (MQCFIN)**
Specifies whether the channel initiator is to be started when the queue manager starts (parameter identifier: **MQIA_CHINIT_CONTROL**).

The value can be:

MQSVC_CONTROL_MANUAL
The channel initiator is not to be started automatically.

MQSVC_CONTROL_Q_MGR
The channel initiator is to be started automatically when the queue manager starts.

This parameter is valid only on IBM i, AIX, Linux, and Windows.

ChannelMonitoring (MQCFIN)
Level of real-time monitoring data collection for channels (parameter identifier: **MQIA_MONITORING_CHANNEL**).

The value can be any of the following values:

MQMON_NONE
Monitoring data collection is disabled, regardless of the setting for the **ChannelMonitoring** channel attribute.

MQMON_OFF
Monitoring data collection is turned off for channels specifying MQMON_Q_MGR in the **ChannelMonitoring** channel attribute.

MQMON_LOW
Monitoring data collection is turned on with a low ratio of data collection for channels specifying MQMON_Q_MGR in the **ChannelMonitoring** channel attribute.

MQMON_MEDIUM
Monitoring data collection is turned on with a moderate ratio of data collection for channels specifying MQMON_Q_MGR in the **ChannelMonitoring** channel attribute.

MQMON_HIGH
Monitoring data collection is turned on with a high ratio of data collection for channels specifying MQMON_Q_MGR in the **ChannelMonitoring** channel attribute.

ChannelStatistics(MQCFIN)
Controls whether statistics data is to be collected for channels (parameter identifier: **MQIA_STATISTICS_CHANNEL**).

The value can be:

MQMON_NONE
Statistics data collection is turned off for channels regardless of the setting of their **ChannelStatistics** parameter. This value is the initial default value of the queue manager.

MQMON_OFF
Statistics data collection is turned off for channels specifying a value of MQMON_Q_MGR in their **ChannelStatistics** parameter.

MQMON_LOW
Statistics data collection is turned on, with a low ratio of data collection, for channels specifying a value of MQMON_Q_MGR in their **ChannelStatistics** parameter.

MQMON_MEDIUM
Statistics data collection is turned on, with a moderate ratio of data collection, for channels specifying a value of MQMON_Q_MGR in their **ChannelStatistics** parameter.

MQMON_HIGH
Statistics data collection is turned on, with a high ratio of data collection, for channels specifying a value of MQMON_Q_MGR in their **ChannelStatistics** parameter.

z/OS On z/OS systems, enabling this parameter simply turns on statistics data collection, regardless of the value you select. Specifying LOW, MEDIUM, or HIGH makes no difference to your results. This parameter must be enabled in order to collect channel accounting records.

ChinitAdapters (MQCFIN)
Number of channel initiator adapter subtasks to use for processing IBM MQ calls (parameter identifier: **MQIA_CHINIT_ADAPTERS**).

This value must be in the range 0 through 9999.

ChinitDispatchers (MQCFIN)
Number of dispatchers to use for the channel initiator (parameter identifier: **MQIA_CHINIT_DISPATCHERS**).

ChinitServiceParm (MQCFST)
This attribute is reserved for use by IBM (parameter identifier: **MQCA_CHINIT_SERVICE_PARM**).

ChinitTraceAutoStart (MQCFIN)
Specifies whether the channel initiator trace should start automatically (parameter identifier: **MQIA_CHINIT_TRACE_AUTO_START**).

The value can be:

MQTRAXSTR_YES
Channel initiator trace starts automatically.

MQTRAXSTR_NO
Channel initiator trace does not starts automatically.

ChinitTraceTableSize (MQCFIN)
Size of the channel initiator's trace data space, in MB (parameter identifier: **MQIA_CHINIT_TRACE_TABLE_SIZE**).

ClusterSenderMonitoring (MQCFIN)
Level of real-time monitoring data collection for auto-defined cluster sender channels (parameter identifier: **MQIA_MONITORING_AUTO_CLUSSDR**).

This parameter can have any of the following values:

MQMON_Q_MGR
The collection of monitoring data is inherited from the setting of the **ChannelMonitoring** attribute in the queue manager object.

MQMON_OFF
Monitoring data collection is disabled.

MQMON_LOW
Monitoring data collection is turned on with a low ratio of data collection.

MQMON_MEDIUM
Monitoring data collection is turned on with a moderate ratio of data collection.

MQMON_HIGH
Monitoring data collection is turned on with a high ratio of data collection.

ClusterSenderStatistics (MQCFIN)
Controls whether statistics data is to be collected for auto-defined cluster-sender channels (parameter identifier: **MQIA_STATISTICS_AUTO_CLUSSDR**).

The value can be:

MQMON_Q_MGR
Collection of statistics data is inherited from the setting of the queue manager's **ChannelStatistics** parameter. This value is the initial default value of the queue manager.

MQMON_OFF
Statistics data collection for the channel is disabled.

MQMON_LOW
Unless **ChannelStatistics** is MQMON_NONE, this value specifies a low rate of data collection with a minimal effect on system performance.

MQMON_MEDIUM
Unless **ChannelStatistics** is MQMON_NONE, this value specifies a moderate rate of data collection.

MQMON_HIGH
Unless **ChannelStatistics** is MQMON_NONE, this value specifies a high rate of data collection.

On z/OS systems, enabling this parameter simply turns on statistics data collection, regardless of the value you select. Specifying LOW, MEDIUM, or HIGH makes no difference to your results. This parameter must be enabled in order to collect channel accounting records.

ClusterWorkLoadData (MQCFST)
Data passed to the cluster workload exit (parameter identifier: **MQCA_CLUSTER_WORKLOAD_DATA**).

ClusterWorkLoadExit (MQCFST)
Name of the cluster workload exit (parameter identifier: **MQCA_CLUSTER_WORKLOAD_EXIT**).

The maximum length of the exit name is MQ_EXIT_NAME_LENGTH.

ClusterWorkLoadLength (MQCFIN)
Cluster workload length (parameter identifier: **MQIA_CLUSTER_WORKLOAD_LENGTH**).

The maximum length of the message passed to the cluster workload exit.

CLWLMRUChannels (MQCFIN)
Maximum number of most recently used channels for cluster workload balancing (parameter identifier: **MQIA_CLWL_MRU_CHANNELS**).

CLWLUseQ (MQCFIN)
This defines the behavior of an MQPUT when the target queue has both a local instance and at least one remote cluster instance (parameter identifier: **MQIA_CLWL_USEQ**).

This parameter can have any of the following values:

MQCLWL_USEQ_ANY
Use remote and local queues.

MQCLWL_USEQ_LOCAL
Do not use remote queues.

CodedCharSetId (MQCFIN)
Coded character set identifier (parameter identifier: **MQIA_CODED_CHAR_SET_ID**).

CommandEvent (MQCFIN)
Controls whether command events are generated (parameter identifier: **MQIA_COMMAND_EVENT**).

The value can be any of the following values:

MQEVR_DISABLED
Command event generation disabled.

MQEVR_ENABLED
Command event generation enabled.

MQEVR_NO_DISPLAY
Command events are generated for all commands other than **MQSC DISPLAY** commands and PCF **Inquire** commands.

CommandEvent (MQCFIN)
Controls whether command events are generated (parameter identifier: **MQIA_COMMAND_EVENT**).

The value can be any of the following values:

MQEVR_DISABLED
Event reporting disabled.

MQEVR_ENABLED
Event reporting enabled.

MQEVR_NO_DISPLAY
Event reporting enabled for all successful commands except Inquire commands.

CommandInputQName (MQCFST)
Command input queue name (parameter identifier: **MQCA_COMMAND_INPUT_Q_NAME**).

The maximum length of the string is MQ_Q_NAME_LENGTH.

CommandLevel (MQCFIN)
Command level supported by queue manager (parameter identifier: **MQIA_COMMAND_LEVEL**).

z/OS **CommandScope (MQCFIN)**
Command scope (parameter identifier: **MQCACF_COMMAND_SCOPE**). This parameter applies to z/OS only.

Specifies how the command is executed when the queue manager is a member of a queue sharing group. You can specify one of the following values:

- Blank (or omit the parameter altogether). The command is executed on the queue manager on which it was entered.
- A queue manager name. The command is executed on the queue manager you specify, providing it is active within the queue sharing group. If you specify a queue manager name other than the queue manager on which it was entered, you must be using a queue sharing group environment. The command server must be enabled.
- An asterisk (*). The command is executed on the local queue manager and is also passed to every active queue manager in the queue sharing group.

The maximum length is MQ_QSG_NAME_LENGTH.

Multi **CommandServerControl (MQCFIN)**
Specifies whether the command server is to be started when the queue manager starts (parameter identifier: **MQIA_CMD_SERVER_CONTROL**).

The value can be:

MQSVC_CONTROL_MANUAL
The command server is not to be started automatically.

MQSVC_CONTROL_Q_MGR
The command server is to be started automatically when the queue manager starts.

This parameter is valid only on IBM i, AIX, Linux, and Windows.

ConfigurationEvent (MQCFIN)
Controls whether configuration events are generated (parameter identifier: **MQIA_CONFIGURATION_EVENT**).

The value can be any of the following values:

MQEVR_DISABLED
Configuration event generation disabled.

MQEVR_ENABLED
Configuration event generation enabled.

ConnAuth (MQCFST)
The name of an authentication information object that is used to provide the location of user ID and password authentication (parameter identifier: **MQCA_CONN_AUTH**).

The maximum length of the string is MQ_AUTH_INFO_NAME_LENGTH. Only authentication information objects with type IDPWOS or IDPWLDAP can be specified; other types result in an error message when the configuration is read by:

- ALW The OAM on AIX, Linux, and Windows.
- z/OS The security component on z/OS

Custom (MQCFST)
Custom attribute for new features (parameter identifier: **MQCA_CUSTOM**).

This attribute is reserved for the configuration of new features before separate attributes are introduced. It can contain the values of zero or more attributes as pairs of attribute name and value, separated by at least one space. The attribute name-value pairs have the form NAME(VALUE). Single quotation marks must be escaped with another single quotation mark.

This description is updated when features using this attribute are introduced. There are no possible values for **Custom**.

The maximum length of the string is MQ_CUSTOM_LENGTH.

CPILevel (MQCFIN)
CPI level (parameter identifier: **MQIA_CPI_LEVEL**).

DeadLetterQName (MQCFST)
Dead letter (undelivered message) queue name (parameter identifier: **MQCA_DEAD_LETTER_Q_NAME**).

Specifies the name of the local queue that is to be used for undelivered messages. Messages are put on this queue if they cannot be routed to their correct destination.

The maximum length of the string is MQ_Q_NAME_LENGTH.

DefXmitQName (MQCFST)
Default transmission queue name (parameter identifier: **MQCA_DEF_XMIT_Q_NAME**).

This is the name of the default transmission queue that is used for the transmission of messages to remote queue managers, if there is no other indication of which transmission queue to use.

The maximum length of the string is MQ_Q_NAME_LENGTH.

DNSGroup (MQCFST)
This parameter is no longer used. From IBM MQ for z/OS 8.0, WLM/DNS is no longer supported by the z/OS Communications Server, so the queue manager attributes **DNSWLM** and **DNSGROUP** are no longer used. (parameter identifier: **MQCA_DNS_GROUP**).

The maximum length of this name is MQ_DNS_GROUP_NAME_LENGTH.

DNSWLM (MQCFIN)
This parameter is no longer used. From IBM MQ for z/OS 8.0, WLM/DNS is no longer supported by the z/OS Communications Server, so the queue manager attributes **DNSWLM** and **DNSGROUP** are no longer used. (parameter identifier: **MQIA_DNS_WLM**).

The value can be any of the following values:

MQDNSWLM_YES
This value may be seen on a queue manager migrated from an earlier release. The value is ignored.

MQDNSWLM_NO
This is the only value supported by the queue manager.

EncryptionPolicySuiteB (MQCFIL)
Specifies whether Suite B-compliant cryptography is used and what level of strength is employed (parameter identifier **MQIA_SUITE_B_STRENGTH**).

The value can be one or more of:

MQ_SUITE_B_NONE
Suite B-compliant cryptography is not used.

MQ_SUITE_B_128_BIT
Suite B 128-bit strength security is used.

MQ_SUITE_B_192_BIT
Suite B 192-bit strength security is used.

If invalid lists are specified, such as MQ_SUITE_B_NONE with MQ_SUITE_B_128_BIT, the error MQRCCF_SUITE_B_ERROR is issued.

ExpiryInterval (MQCFIN)
Expiry interval (parameter identifier: **MQIA_EXPIRY_INTERVAL**).

Force (MQCFIN)
Force changes (parameter identifier: **MQIACF_FORCE**).

Specifies whether the command is forced to complete if both of the following are true:

- **DefXmitQName** is specified, and
- An application has a remote queue open, the resolution for which is affected by this change.

z/OS **GroupUR (MQCFIN)**
Controls whether XA client applications can establish transactions with a GROUP unit of recovery disposition (parameter identifier: **MQIA_GROUP_UR**).

The value can be any of the following values:

MQGUR_DISABLED
XA client applications must connect using a queue manager name.

MQGUR_ENABLED
XA client applications can establish transactions with a group unit of recovery disposition by specifying a queue sharing group name when they connect.

z/OS **IGQPutAuthority (MQCFIN)**
IGQ put authority (parameter identifier: **MQIA_IGQ_PUT_AUTHORITY**).

z/OS **IGQUserId (MQCFST)**
Intra-group queuing agent user identifier (parameter identifier: **MQCA_IGQ_USER_ID**). This parameter is valid only on z/OS when the queue manager is a member of a queue sharing group.

Specifies the user identifier that is associated with the local intra-group queuing agent. This identifier is one of the user identifiers that might be checked for authorization when the IGQ agent puts messages on local queues. The actual user identifiers checked depend on the setting of the **IGQPutAuthority** attribute, and on external security options.

The maximum length is MQ_USER_ID_LENGTH.

Multi **ImageInterval (MQCFIN)**
The target frequency with which the queue manager automatically writes media images, in minutes since the previous media image for an object (parameter identifier: **MQIA_MEDIA_IMAGE_INTERVAL**). This parameter is not valid on z/OS.

The value can be:

Time interval
The time in minutes from 1 - 999 999 999, at which the queue manager automatically writes media images.

The default value is 60 minutes.

MQMEDIMGINTVL_OFF
Automatic media images are not written on a time interval basis.

Multi **ImageLogLength (MQCFIN)**
The target size of the recovery log, written before the queue manager automatically writes media images, in number of megabytes since the previous media image for an object. This limits the amount of log to be read when recovering an object (parameter identifier: **MQIA_MEDIA_IMAGE_LOG_LENGTH**). This parameter is not valid on z/OS.

The value can be:

Target log size
The target size of the recovery log in megabytes from 1 - 999 999 999.

MQMEDIMGLOGLN_OFF
Automatic media images are not written based on the size of log written.

MQMEDIMGLOGLN_OFF is the default value.

Multi **ImageRecoverObject (MQCFST)**
Specifies whether authentication information, channel, client connection, listener, namelist, process, alias queue, remote queue, and service objects are recoverable from a media image, if linear logging is being used (parameter identifier: **MQIA_MEDIA_IMAGE_RECOVER_OBJ**). This parameter is not valid on z/OS.

The value can be:

MQIMGRCOV_NO
The rcdmqimg and rcrmqobj commands are not permitted for these objects, and automatic media images, if enabled, are not written for these objects.

MQIMGRCOV_YES
These objects are recoverable.

MQIMGRCOV_YES is the default value.

Multi **ImageRecoverObject (MQCFST)**
Specifies whether authentication information, channel, client connection, listener, namelist, process, alias queue, remote queue, and service objects are recoverable from a media image, if linear logging is being used (parameter identifier: **MQIA_MEDIA_IMAGE_RECOVER_OBJ**). This parameter is not valid on z/OS.

The value can be:

MQIMGRCOV_NO
The rcdmqimg and rcrmqobj commands are not permitted for these objects, and automatic media images, if enabled, are not written for these objects.

MQIMGRCOV_YES
These objects are recoverable.

MQIMGRCOV_YES is the default value.

Multi **ImageRecoverQueue (MQCFST)**
Specifies the default **ImageRecoverQueue** attribute for local and permanent dynamic queue objects, when used with this parameter (parameter identifier: **MQIA_MEDIA_IMAGE_RECOVER_Q**). This parameter is not valid on z/OS.

The value can be:

MQIMGRCOV_NO
The **ImageRecoverQueue** attribute for local and permanent dynamic queue objects is set to MQIMGRCOV_NO .

MQIMGRCOV_YES
The **ImageRecoverQueue** attribute for local and permanent dynamic queue objects is set to MQIMGRCOV_YES .

MQIMGRCOV_YES is the default value.

Multi **ImageSchedule (MQCFST)**
Whether the queue manager automatically writes media images (parameter identifier: **MQIA_MEDIA_IMAGE_SCHEDULING**). This parameter is not valid on z/OS.

The value can be:

MQMEDIMGSCHED_AUTO
The queue manager attempts to automatically write a media image for an object, before **ImageInterval** minutes have elapsed, or **ImageLogLength** megabytes of recovery log have been written, since the previous media image for the object was taken.

The previous media image might have been taken manually or automatically, depending on the settings of **ImageInterval** or **ImageLogLength**.

MQMEDIMGSCHED_MANUAL
Automatic media images are not written.

MQMEDIMGSCHED_MANUAL is the default value.

InhibitEvent (MQCFIN)
Controls whether inhibit (Inhibit Get and Inhibit Put) events are generated (parameter identifier: **MQIA_INHIBIT_EVENT**).

The value can be:

MQEVR_DISABLED
Event reporting disabled.

MQEVR_ENABLED
Event reporting enabled.

z/OS **IntraGroupQueuing (MQCFIN)**
Intra group queuing (parameter identifier: **MQIA_INTRA_GROUP_QUEUING**).

IPAddressVersion (MQCFIN)
Specifies the IP version to be used (parameter identifier: **MQIA_IP_ADDRESS_VERSION**).

The value can be any of the following values:

MQIPADDR_IPV4
The IPv4 stack is used.

MQIPADDR_IPV6
The IPv6 stack is used.

ListenerTimer (MQCFIN)
The time interval, in seconds, between attempts to restart a listener following an APPC or TCP/IP failure (parameter identifier: **MQCA_LISTENER_TIMER**).

LocalEvent (MQCFIN)
Controls whether local error events are generated (parameter identifier: **MQIA_LOCAL_EVENT**).

The value can be any of the following values:

MQEVR_DISABLED
Event reporting disabled.

MQEVR_ENABLED
Event reporting enabled.

Multi **LoggerEvent (MQCFIN)**
Controls whether recovery log events are generated (parameter identifier: **MQIA_LOGGER_EVENT**).

The value can be:

MQEVR_DISABLED
Event reporting disabled.

MQEVR_ENABLED
Event reporting enabled. This value is valid only on queue managers that use linear logging.

This parameter is valid only on IBM i, AIX, Linux, and Windows.

z/OS **LUGroupName (MQCFST)**
Generic LU name for the LU 6.2 listener (parameter identifier: **MQCA_LU_GROUP_NAME**).

The generic LU name to be used by the LU 6.2 listener that handles inbound transmissions for the queue sharing group.

This parameter applies to z/OS only.

The maximum length of the string is MQ_LU_NAME_LENGTH.

z/OS **LUName (MQCFST)**
LU name to use for outbound LU 6.2 transmissions (parameter identifier: **MQCA_LU_NAME**).

The name of the LU to use for outbound LU 6.2 transmissions. Set this parameter to be the same as the name of the LU to be used by the listener for inbound transmissions.

This parameter applies to z/OS only.

The maximum length of the string is MQ_LU_NAME_LENGTH.

LU62ARMSuffix (MQCFST)
The suffix of the SYS1.PARMLIB member APPCPMxx, that nominates the LUADD for this channel initiator (parameter identifier: MQCA_LU62_ARM_SUFFIX).

The maximum length of this name is MQ_ARM_SUFFIX_LENGTH.

LU62Channels (MQCFIN)
Maximum number of current channels that use the LU 6.2 transmission protocol, including clients connected to server connection channels (parameter identifier: **MQIA_LU62_CHANNELS**).

LUGroupName (MQCFST)
The generic LU name that the LU 6.2 listener that handles inbound transmissions for the queue sharing group is to use. This name must be the same as **LUName** (parameter identifier: **MQCA_LU_GROUP_NAME**).

The maximum length of this name is MQ_LU_NAME_LENGTH.

LUName (MQCFST)
The LU name that the LU 6.2 listener that handles outbound transmissions is to use. This name must be the same as **LUGroupName** (parameter identifier: **MQCA_LU_NAME**).

The maximum length of this name is MQ_LU_NAME_LENGTH.

MaxActiveChannels (MQCFIN)
Maximum number of channels that can be active at the same time (parameter identifier: **MQIA_ACTIVE_CHANNELS**).

MaxChannels (MQCFIN)
Maximum number of current channels, including clients connected to server connection channels (parameter identifier: **MQIA_MAX_CHANNELS**).

MaxHandles (MQCFIN)
Maximum number of handles (parameter identifier: **MQIA_MAX_HANDLES**).

Specifies the maximum number of handles that any one job can have open at the same time.

MaxMsgLength (MQCFIN)
Maximum message length (parameter identifier: **MQIA_MAX_MSG_LENGTH**).

MaxPriority (MQCFIN)
Maximum priority (parameter identifier: MQIA_MAX_PRIORITY).

MaxUncommittedMsgs (MQCFIN)
Maximum number of uncommitted messages within a unit of work (parameter identifier: **MQIA_MAX_UNCOMMITTED_MSGS**).

That is:

- The number of messages that can be retrieved, plus
- The number of messages that can be put on a queue, plus
- Any trigger messages generated within this unit of work

under any one syncpoint. This limit does not apply to messages that are retrieved or put outside syncpoint.

MQIAccounting(MQCFIN
Controls whether accounting information for MQI data is to be collected (parameter identifier: **MQIA_ACCOUNTING_MQI**).

The value can be:

MQMON_OFF
MQI accounting data collection is disabled. This value is the initial default value of the queue manager.

MQMON_ON
MQI accounting data collection is enabled.

This parameter is valid only on IBM i, AIX, Linux, and Windows.

MQIStatistics(MQCFIN)
Controls whether statistics monitoring data is to be collected for the queue manager (parameter identifier: **MQIA_STATISTICS_MQI**).

The value can be:

MQMON_OFF
Data collection for MQI statistics is disabled. This value is the initial default value of the queue manager.

MQMON_ON
Data collection for MQI statistics is enabled.

This parameter is valid only on IBM i, AIX, Linux, and Windows.

MsgMarkBrowseInterval(MQCFIN)
Mark-browse interval (parameter identifier: **MQIA_MSG_MARK_BROWSE_INTERVAL**).

Specifies the time interval in milliseconds after which the queue manager can automatically unmark messages.

This parameter can either have a value in the range 0 - 999,999,999, or the special value `MQMMBI_UNLIMITED`.

A value of `0` causes the queue manager to unmark messages immediately.

`MQMMBI_UNLIMITED` indicates that the queue manager does not automatically unmark messages.

OutboundPortMax (MQCFIN)
Outbound port range maximum (parameter identifier: **MQIA_OUTBOUND_PORT_MAX**).

The upper limit for the range of port numbers used when binding outgoing channels.

OutboundPortMin (MQCFIN)
Outbound port range minimum (parameter identifier: **MQIA_OUTBOUND_PORT_MIN**).

The lower limit for the range of port numbers used when binding outgoing channels.

Parent(MQCFST)
The name of the queue manager to which this queue manager is to connect hierarchically as its child (parameter identifier: **MQCA_PARENT**).

A blank value indicates that this queue manager has no parent queue manager. If there is an existing parent queue manager it is disconnected. This value is the initial default value of the queue manager.

The maximum length of the string is MQ_Q_MGR_NAME_LENGTH.

Note:

- The use of IBM MQ hierarchical connections requires that the queue manager attribute **PSMode** is set to MQPSM_ENABLED.
- The value of **Parent** can be set to a blank value if **PSMode** is set to MQPSM_DISABLED.
- Before connecting to a queue manager hierarchically as its child, channels in both directions must exist between the parent queue manager and child queue manager.
- If a parent is defined, the **Change Queue Manager** command disconnects from the original parent and sends a connection flow to the new parent queue manager.
- Successful completion of the command does not mean that the action completed or that it is going to complete successfully. Use the **Inquire Pub/Sub Status** command to track the status of the requested parent relationship.

PerformanceEvent (MQCFIN)
Controls whether performance-related events are generated (parameter identifier: **MQIA_PERFORMANCE_EVENT**).

The value can be any of the following values:

MQEVR_DISABLED
Event reporting disabled.

MQEVR_ENABLED
Event reporting enabled.

Platform (MQCFIN)
Platform on which the queue manager resides (parameter identifier: **MQIA_PLATFORM**).

PubSubClus (MQCFIN)
Controls whether the queue manager participates in publish/subscribe clustering (parameter identifier: **MQIA_PUBSUB_CLUSTER**).

The value can be:

MQPSCLUS_ENABLED
The creating or receipt of clustered topic definitions and cluster subscriptions is permitted.

Note: The introduction of a clustered topic into a large IBM MQ cluster can cause a degradation in performance. This degradation occurs because all partial repositories are notified of all the other members of the cluster. Unexpected subscriptions might be created at all other nodes; for example; where **proxysub(FORCE)** is specified. Large numbers of channels might be started from a queue manager; for example, on resync after a queue manager failure.

MQPSCLUS_DISABLED
The creating or receipt of clustered topic definitions and cluster subscriptions is inhibited. The creations or receipts are recorded as warnings in the queue manager error logs.

PubSubMaxMsgRetryCount (MQCFIN)
The number of attempts to reprocess a message when processing a failed command message under sync point (parameter identifier: **MQIA_PUBSUB_MAXMSG_RETRY_COUNT**).

The value of this parameter must be a number in the range 0 to 999 999 999. The initial value is 5.

PubSubMode (MQCFIN)
Specifies whether the publish/subscribe engine and the queued publish/subscribe interface are running. The publish/subscribe engine enables applications to publish or subscribe by using the

application programming interface. The publish/subscribe interface monitors the queues used the queued publish/subscribe interface (parameter identifier: **MQIA_PUBSUB_MODE**).

The value can be:

MQPSM_COMPAT
The publish/subscribe engine is running. It is therefore possible to publish or subscribe by using the application programming interface. The queued publish/subscribe interface is not running. Therefore any message that is put to the queues that are monitored by the queued publish/subscribe interface is not acted on. MQPSM_COMPAT is used for compatibility with versions of IBM Integration Bus (formerly known as WebSphere Message Broker) prior to version 7 that use this queue manager.

MQPSM_DISABLED
The publish/subscribe engine and the queued publish/subscribe interface are not running. It is therefore not possible to publish or subscribe using the application programming interface. Any publish/subscribe messages that are put to the queues that are monitored by the queued publish/subscribe interface are not acted on.

MQPSM_ENABLED
The publish/subscribe engine and the queued publish/subscribe interface are running. It is therefore possible to publish or subscribe by using the application programming interface and the queues that are monitored by the queued publish/subscribe interface. This value is the initial default value of the queue manager.

PubSubNPInputMsg (MQCFIN)
Whether to discard (or keep) an undelivered input message (parameter identifier: **MQIA_PUBSUB_NP_MSG**).

The value can be:

MQUNDELIVERED_DISCARD
Non-persistent input messages are discarded if they cannot be processed.

MQUNDELIVERED_KEEP
Non-persistent input messages are not discarded if they cannot be processed. In this situation, the queued publish/subscribe interface continues to try the process again at appropriate intervals and does not continue processing subsequent messages.

PubSubNPResponse (MQCFIN)
Controls the behavior of undelivered response messages (parameter identifier: **MQIA_PUBSUB_NP_RESP**).

The value can be:

MQUNDELIVERED_NORMAL
Non-persistent responses that cannot be placed on the reply queue are put on the dead letter queue. If they cannot be placed on the dead letter queue they are discarded.

MQUNDELIVERED_SAFE
Non-persistent responses that cannot be placed on the reply queue are put on the dead letter queue. If the response cannot be sent and cannot be placed on the dead letter queue the queued publish/subscribe interface rolls back the current operation. The operation is tried again at appropriate intervals and does not continue processing subsequent messages.

MQUNDELIVERED_DISCARD
Non-persistent responses that are not placed on the reply queue are discarded.

MQUNDELIVERED_KEEP
Non-persistent responses are not placed on the dead letter queue or discarded. Instead, the queued publish/subscribe interface backs out the current operation and then try it again at appropriate intervals.

PubSubSyncPoint (MQCFIN)
Whether only persistent (or all) messages must be processed under sync point (parameter identifier: **MQIA_PUBSUB_SYNC_PT**).

The value can be:

MQSYNCPOINT_IFPER
This value makes the queued publish/subscribe interface receive non-persistent messages outside sync point. If the interface receives a publication outside sync point, the interface forwards the publication to subscribers known to it outside sync point.

MQSYNCPOINT_YES
This value makes the queued publish/subscribe interface receive all messages under sync point.

QMgrDesc (MQCFST)
Queue manager description (parameter identifier: **MQCA_Q_MGR_DESC**).

The maximum length of the string is MQ_Q_MGR_DESC_LENGTH.

QMgrIdentifier (MQCFST)
Queue manager identifier (parameter identifier: **MQCA_Q_MGR_IDENTIFIER**).

The unique identifier of the queue manager.

QMgrName (MQCFST)
Name of local queue manager (parameter identifier: **MQCA_Q_MGR_NAME**).

The maximum length of the string is MQ_Q_MGR_NAME_LENGTH.

QSGName (MQCFST)
Queue sharing group name (parameter identifier: **MQCA_QSG_NAME**).

The maximum length of the string is MQ_QSG_NAME_LENGTH.

z/OS **QSGCertificateLabel (MQCFST)**
Specifies the certificate label for the queue sharing group to use (parameter identifier: **MQCA_QSG_CERT_LABEL**).

QueueAccounting (MQCFIN)
Specifies whether accounting information is collected for queues (parameter identifier: **MQIA_ACCOUNTING_Q**).

The value can be any of the following values:

MQMON_ON
For all queues that have the queue parameter **QueueAccounting** specified as MQMON_Q_MGR, accounting information is collected.

MQMON_OFF
For all queues that have the queue parameter **QueueAccounting** specified as MQMON_Q_MGR, accounting information is not collected.

MQMON_NONE
Accounting information is not collected for queues.

QueueMonitoring (MQCFIN)
Level of real-time monitoring data collection for queues (parameter identifier: **MQIA_MONITORING_Q**).

The value can be any of the following values:

MQMON_NONE
Monitoring data collection is disabled, regardless of the setting for the **QueueMonitoring** queue attribute.

MQMON_OFF
Monitoring data collection is turned off for queues specifying MQMON_Q_MGR in the **QueueMonitoring** queue attribute.

MQMON_LOW
Monitoring data collection is turned on with a low ratio of data collection for queues specifying MQMON_Q_MGR in the **QueueMonitoring** queue attribute.

MQMON_MEDIUM
Monitoring data collection is turned on with a moderate ratio of data collection for queues specifying MQMON_Q_MGR in the **QueueMonitoring** queue attribute.

MQMON_HIGH
Monitoring data collection is turned on with a high ratio of data collection for queues specifying MQMON_Q_MGR in the **QueueMonitoring** queue attribute.

Multi **QueueStatistics (MQCFIN)**
Controls whether statistics data is to be collected for queues (parameter identifier: **MQIA_STATISTICS_Q**).

The value can be:

MQMON_NONE
Statistics data collection is turned off for queues regardless of the setting of their **QueueStatistics** parameter. This value is the initial default value of the queue manager.

MQMON_OFF
Statistics data collection is turned off for queues specifying a value of MQMON_Q_MGR in their **QueueStatistics** parameter.

MQMON_ON
Statistics data collection is turned on for queues specifying a value of MQMON_Q_MGR in their **QueueStatistics** parameter.

This parameter is valid only on IBM i, AIX, Linux, and Windows.

ReceiveTimeout(MQCFIN)
In conjunction with **ReceiveTimeoutType** specifies how long a TCP/IP channel will wait to receive data, including heartbeats, from its partner before returning to the inactive state (parameter identifier: **MQIA_RECEIVE_TIMEOUT**).

ReceiveTimeoutMin(MQCFIN)
The minimum time, in seconds, that a TCP/IP channel will wait to receive data, including heartbeats, from its partner before returning to the inactive state (parameter identifier: **MQIA_RECEIVE_TIMEOUT_MIN**).

ReceiveTimeoutType (MQCFIN)
In conjunction with **ReceiveTimeout** specifies how long a TCP/IP channel will wait to receive data, including heartbeats, from its partner before returning to the inactive state (parameter identifier: **MQIA_RECEIVE_TIMEOUT_TYPE**).

The value can be any of the following values:

MQRCVTIME_MULTIPLY
The **ReceiveTimeout** value is a multiplier to be applied to the negotiated value of **HeartbeatInterval** to determine how long a channel will wait. This is the queue manager's initial default value.

MQRCVTIME_ADD
ReceiveTimeout is a value, in seconds, to be added to the negotiated value of **HeartbeatInterval** to determine how long a channel will wait.

MQRCVTIME_EQUAL
ReceiveTimeout is a value, in seconds, representing how long a channel will wait.

RemoteEvent (MQCFIN)
Controls whether remote error events are generated (parameter identifier: **MQIA_REMOTE_EVENT**).

The value can be any of the following values:

MQEVR_DISABLED
Event reporting disabled.

MQEVR_ENABLED
Event reporting enabled.

RepositoryName (MQCFST)
Repository name (parameter identifier: **MQCA_REPOSITORY_NAME**).

The name of a cluster for which this queue manager is to provide a repository service.

RepositoryNamelist (MQCFST)
Repository name list (parameter identifier: **MQCA_REPOSITORY_NAMELIST**).

The name of a list of clusters for which this queue manager is to provide a repository service.

RevDns (MQCFIN)
Whether reverse lookup of the host name from a Domain Name Server is carried out. (parameter identifier: **MQIA_REVERSE_DNS_LOOKUP**).

This attribute has an effect only on channels using a transport type (TRPTYPE) of TCP.

The value can be:

MQRDNS_DISABLED
DNS host names are not reverse looked-up for the IP addresses of inbound channels. With this setting any CHLAUTH rules using host names are not matched.

MQRDNS_ENABLED
DNS host names are reverse looked-up for the IP addresses of inbound channels when this information is required. This setting is required for matching against CHLAUTH rules that contain host names, and for writing out error messages.

z/OS **SecurityCase (MQCFIN)**
Security case supported (parameter identifier: **MQIA_SECURITY_CASE**).

Specifies whether the queue manager supports security profile names in mixed case, or in uppercase only. The value is activated when a Refresh Security command is run with **SecurityType(MQSECTYPE_CLASSES)** specified. This parameter is valid only on z/OS.

The value can be:

MQSCYC_UPPER
Security profile names must be in uppercase.

MQSCYC_MIXED
Security profile names can be in uppercase or in mixed case.

z/OS **SharedQueueQueueManagerName (MQCFIN)**
Specifies how messages are put on a shared queue that specifies another queue manager from a queue sharing group as the object queue manager (parameter identifier: **MQIA_SHARED_Q_Q_MGR_NAME**).

The value can be:

MQSQQM_USE
Messages are delivered to the object queue manager before being put on the shared queue.

MQSQQM_IGNORE
Messages are put directly on the shared queue.

SSLCRLNameList (MQCFST)
TLS CRL name list (parameter identifier: **MQCA_SSL_CRL_NAMELIST**).

The maximum length of the string is `MQ_NAMELIST_NAME_LENGTH`.

SSLEvent (MQCFIN)
Determines whether IMS bridge events are generated (parameter identifier: **MQIA_SSL_EVENT**).

The value can be any of the following values:

MQEVR_ENABLED
All TLS events are enabled.

MQEVR_DISABLED
All TLS events are disabled.

ALW **SSLCryptoHardware(MQCFST)**
The TLS cryptographic hardware (parameter identifier: **MQCA_SSL_CRYPTO_HARDWARE**).

The length of the string is MQ_SSL_CRYPTO_HARDWARE_LENGTH.

Sets the name of the parameter string required to configure the cryptographic hardware present on the system.

This parameter is valid only on AIX, Linux, and Windows.

All supported cryptographic hardware supports the PKCS #11 interface. Specify a string of the following format:

```
GSK_PKCS11=PKCS_#11_driver_path_and_filename;PKCS_#11_token_label;PKCS_#11_token_password;sy
mmetric_cipher_setting;
```

The PKCS #11 driver path is an absolute path to the shared library providing support for the PKCS #11 card. The PKCS #11 driver filename is the name of the shared library. An example of the value required for the PKCS #11 driver path and filename is `/usr/lib/pkcs11/PKCS11_API.so`.

To access symmetric cipher operations through IBM Global Security Kit (GSKit), specify the symmetric cipher setting parameter. The value of this parameter is either:

SYMMETRIC_CIPHER_OFF
Do not access symmetric cipher operations.

SYMMETRIC_CIPHER_ON
Access symmetric cipher operations.

If the symmetric cipher setting is not specified, this value has the same effect as specifying SYMMETRIC_CIPHER_OFF.

The maximum length of the string is 256 characters. The default value is blank.

If you specify a string in the wrong format, you get an error.

When the **SSLCryptoHardware (MQCFST)** value is changed, the cryptographic hardware parameters specified become the ones used for new TLS connection environments. The new information becomes effective:

- When a new channel process is started.
- For channels that run as threads of the channel initiator, when the channel initiator is restarted.
- For channels that run as threads of the listener, when the listener is restarted.
- When a Refresh Security command is issued to refresh the contents of the TLS key repository.

SSLEvent (MQCFIN)
Controls whether TLS events are generated (parameter identifier: **MQIA_SSL_EVENT**).

The value can be:

MQEVR_DISABLED
Event reporting disabled.

MQEVR_ENABLED
Event reporting enabled.

z/OS ALW **SSLFipsRequired (MQCFIN)**
SSLFIPS specifies whether only FIPS certified algorithms are to be used if cryptography is carried out in IBM MQ, rather than in cryptographic hardware (parameter identifier: **MQIA_SSL_FIPS_REQUIRED**).

If cryptographic hardware is configured, the cryptographic modules used are those modules provided by the hardware product. These modules might, or might not, be FIPS certified to a particular level

depending on the hardware product in use. This parameter applies to z/OS, AIX, Linux, and Windows platforms only.

The value can be any of the following values:

MQSSL_FIPS_NO
IBM MQ provides an implementation of TLS cryptography which supplies some FIPS certified modules on some platforms. If you set **SSLFIPSRequired** to MQSSL_FIPS_NO, any CipherSpec supported on a particular platform can be used. This value is the initial default value of the queue manager.

If the queue manager runs without using cryptographic hardware, refer to the CipherSpecs listed in Specifying CipherSpecs employing FIPS 140-2 certified cryptography:

MQSSL_FIPS_YES
Specifies that only FIPS certified algorithms are to be used in the CipherSpecs allowed on all TLS connections from and to this queue manager.

For a listing of appropriate FIPS 140-2 certified CipherSpecs; see Specifying CipherSpecs.

Changes to **SSLFIPS** become effective in the following cases:

- ALW On AIX, Linux, and Windows, when a new channel process is started.
- ALW For channels that run as threads of the channel initiator on AIX, Linux, and Windows, when the channel initiator is restarted.
- ALW For channels that run as threads of the listener on AIX, Linux, and Windows, when the listener is restarted.
- ALW For channels that run as threads of a process pooling process, when the process pooling process is started or restarted and first runs a TLS channel. If the process pooling process has already run a TLS channel, and you want the change to become effective immediately, run the MQSC command **REFRESH SECURITY TYPE(SSL)**. The process pooling process is **amqrmppa** on AIX, Linux, and Windows.
- z/OS On z/OS, when the channel initiator is restarted.
- z/OS When a **REFRESH SECURITY TYPE(SSL)** command is issued, except on z/OS.

SSLKeyRepository (MQCFST)
TLS key repository (parameter identifier: **MQCA_SSL_KEY_REPOSITORY**).

The maximum length of the string is MQ_SSL_KEY_REPOSITORY_LENGTH.

SSLKeyResetCount (MQCFIN)
TLS key reset count (parameter identifier: **MQIA_SSL_RESET_COUNT**).

The maximum length of the string is MQ_SSL_KEY_REPOSITORY_LENGTH.

SSLTasks (MQCFIN)
TLS tasks (parameter identifier: **MQIA_SSL_TASKS**).

StartStopEvent (MQCFIN)
Controls whether start and stop events are generated (parameter identifier: **MQIA_START_STOP_EVENT**).

The value can be any of the following values:

MQEVR_DISABLED
Event reporting disabled.

MQEVR_ENABLED
Event reporting enabled.

Multi **StatisticsInterval (MQCFIN)**
The time interval, in seconds, at which statistics monitoring data is written to the monitoring queue (parameter identifier: **MQIA_STATISTICS_INTERVAL**).

Specify a value in the range 1 - 604,000.

This parameter is valid only on IBM i, AIX, Linux, and Windows.

SyncPoint (MQCFIN)
Syncpoint availability (parameter identifier: **MQIA_SYNCPOINT**).

TCPChannels (MQCFIN)
Maximum number of current channels that use the TCP/IP transmission protocol, including clients connected to server connection channels (parameter identifier: **MQIA_TCP_CHANNELS**).

TCPKeepAlive (MQCFIN)
Specifies whether to use the TCP KEEPALIVE facility to check whether the MCA at the opposite end of a channel is available (parameter identifier: **MQIA_TCP_KEEP_ALIVE**).

The value can be any of the following values:

MQTCPKEEP_YES
Use the TCP KEEPALIVE facility as specified in the TCP profile configuration data set.

MQTCPKEEP_NO
Do not use the TCP KEEPALIVE facility.

TCPName (MQCFST)
TCP name (parameter identifier: **MQIA_TCP_NAME**).

The name of the current TCP/IP system in use.

The maximum length of this value is MQ_TCP_NAME_LENGTH.

TCPStackType (MQCFIN)
TCP stack type (parameter identifier: **MQIA_TCP_STACK_TYPE**).

Specifies whether the channel initiator uses the TCP/IP address space specified in TCPNAME only, or whether it can bind to any selected TCP/IP address.

The value can be:

MQTCPSTACK_SINGLE
The channel initiator uses the TCP/IP address space specified in TCPNAME only.

MQTCPSTACK_MULTIPLE
The initiator can use any TCP/IP address space available to it. If no other address spaces are available, the address space specified in TCPNAME is used.

TraceRouteRecording (MQCFIN)
Specifies whether trace-route messaging is enabled or disabled (parameter identifier: **MQIA_TRACE_ROUTE_RECORDING**).

The value can be:

MQRECORDING_MSG
Trace-route messaging is enabled. Trace-route reply messages are delivered to the reply-to queue specified in the message descriptor of the message.

MQRECORDING_Q
Trace-route messaging is enabled. Trace-route reply messages are delivered to a fixed name queue.

MQRECORDING_DISABLED.
Trace-route messaging is disabled.

TreeLifeTime (MQCFIN)
The lifetime, in seconds, of non-administrative topics (parameter identifier: **MQIA_TREE_LIFE_TIME**).

Non-administrative topics are those topics created when an application publishes to, or subscribes as, a topic string that does not exist as an administrative node. When this non-administrative node no longer has any active subscriptions, this parameter determines how long the queue manager waits before removing that node. Only non-administrative topics that are in use by a durable subscription remain after the queue manager is recycled.

Specify a value in the range 0 - 604,000. A value of 0 means that non-administrative topics are not removed by the queue manager. The initial default value of the queue manager is 1800.

TriggerInterval (MQCFIN)
Trigger interval (parameter identifier: **MQIA_TRIGGER_INTERVAL**).

Specifies the trigger time interval, expressed in milliseconds, for use only with queues where TriggerType has a value of MQTT_FIRST.

Storage class attributes

Event messages relating to objects can include storage class attributes

AlterationDate (MQCFST)
Alteration date (parameter identifier: **MQCA_ALTERATION_DATE**).

The date when the information was last altered.

AlterationTime (MQCFST)
Alteration time (parameter identifier: **MQCA_ALTERATION_TIME**).

The time when the information was last altered.

PageSetId (MQCFIN)
Page set identifier (parameter identifier: **MQIA_PAGESET_ID**).

PassTicketApplication (MQCFST)
Name of the application used to authenticate IMS bridge passtickets (parameter identifier: **MQCA_PASS_TICKET_APPL**).

The maximum length of the string is MQ_PASS_TICKET_APPL_LENGTH.

StgClassDesc (MQCFST)
Storage class description (parameter identifier: **MQCA_STORAGE_CLASS_DESC**).

The maximum length of the string is MQ_STORAGE_CLASS_DESC_LENGTH.

XCFGroupName (MQCFST)
XCF group name (parameter identifier: **MQCA_XCF_GROUP_NAME**).

The maximum length of the string is MQ_XCF_GROUP_NAME_LENGTH.

XCFMemberName (MQCFST)
XCF member name (parameter identifier: MQCA_XCF_MEMBER_NAME).

The maximum length of the string is **MQ_XCF_MEMBER_NAME_LENGTH**.

Topic attributes

Event messages relating to objects can include topic attributes

AlterationDate (MQCFST)
Alteration date (parameter identifier: **MQCA_ALTERATION_DATE**).

The date when the information was last altered, in the form *yyyy-mm-dd*.

AlterationTime (MQCFST)
Alteration time (parameter identifier: **MQCA_ALTERATION_TIME**).

The time when the information was last altered, in the form *hh.mm.ss*.

ClusterName (MQCFST)
The name of the cluster to which this topic belongs. (parameter identifier: **MQCA_CLUSTER_NAME**).

The maximum length of the string is MQ_CLUSTER_NAME_LENGTH. Setting this parameter to a cluster that this queue manager is a member of makes all queue managers in the cluster aware of this topic. Any publication to this topic or a topic string below it put to any queue manager in the cluster is propagated to subscriptions on any other queue manager in the cluster. For more details, see Distributed publish/subscribe networks.

The value can be any of the following values:

Blank
If no topic object above this topic in the topic tree has set this parameter to a cluster name, then this topic does not belong to a cluster. Publications and subscriptions for this topic are not propagated to publish/subscribe cluster-connected queue managers. If a topic node higher in the topic tree has a cluster name set, publications and subscriptions to this topic are also propagated throughout the cluster.

This value is the default value for this parameter if no value is specified.

String
The topic belongs to this cluster. It is not recommended that this is set to a different cluster from a topic object above this topic object in the topic tree. Other queue managers in the cluster will honor this object's definition unless a local definition of the same name exists on those queue managers.

Additionally, if **PublicationScope** or **SubscriptionScope** are set to MQSCOPE_ALL, this value is the cluster to be used for the propagation of publications and subscriptions, for this topic, to publish/subscribe cluster-connected queue managers.

DefPersistence (MQCFIN)
Default persistence (parameter identifier: **MQIA_TOPIC_DEF_PERSISTENCE**).

The value can be:

MQPER_PERSISTENCE_AS_PARENT
The default persistence is based on the setting of the closest parent administrative topic object in the topic tree.

MQPER_PERSISTENT
Message is persistent.

MQPER_NOT_PERSISTENT
Message is not persistent.

DefPriority (MQCFIN)
Default priority (parameter identifier: **MQIA_DEF_PRIORITY**).

DefPutResponse (MQCFIN)
Default put response (parameter identifier: **MQIA_DEF_PUT_RESPONSE_TYPE**).

The value can be:

MQPRT_ASYNC_RESPONSE
The put operation is issued asynchronously, returning a subset of MQMD fields.

MQPRT_RESPONSE_AS_PARENT
The default put response is based on the setting of the closest parent administrative topic object in the topic tree.

MQPRT_SYNC_RESPONSE
The put operation is issued synchronously, returning a response.

DurableModelQName (MQCFST)
Name of the model queue to be used for durable managed subscriptions (parameter identifier: **MQCA_MODEL_DURABLE_Q**).

The maximum length of the string is MQ_Q_NAME_LENGTH.

DurableSubscriptions (MQCFIN)
Whether applications are permitted to make durable subscriptions (parameter identifier: **MQIA_DURABLE_SUB**).

The value can be:

MQSUB_DURABLE_AS_PARENT
Whether durable subscriptions are permitted is based on the setting of the closest parent administrative topic object in the topic tree.

MQSUB_DURABLE_ALLOWED
Durable subscriptions are permitted.

MQSUB_DURABLE_INHIBITED
Durable subscriptions are not permitted.

InhibitPublications (MQCFIN)
Whether publications are allowed for this topic (parameter identifier: **MQIA_INHIBIT_PUB**).

The value can be:

MQTA_PUB_AS_PARENT
Whether messages can be published to this topic is based on the setting of the closest parent administrative topic object in the topic tree.

MQTA_PUB_INHIBITED
Publications are inhibited for this topic.

MQTA_PUB_ALLOWED
Publications are allowed for this topic.

InhibitSubscriptions (MQCFIN)
Whether subscriptions are allowed for this topic (parameter identifier: **MQIA_INHIBIT_SUB**).

The value can be:

MQTA_SUB_AS_PARENT
Whether applications can subscribe to this topic is based on the setting of the closest parent administrative topic object in the topic tree.

MQTA_SUB_INHIBITED
Subscriptions are inhibited for this topic.

MQTA_SUB_ALLOWED
Subscriptions are allowed for this topic.

NonDurableModelQName (MQCFST)
Name of the model queue to be used for non durable managed subscriptions (parameter identifier: **MQCA_MODEL_NON_DURABLE_Q**).

The maximum length of the string is MQ_Q_NAME_LENGTH.

NonPersistentMsgDelivery (MQCFIN)
The delivery mechanism for non-persistent messages published to this topic (parameter identifier: **MQIA_NPM_DELIVERY**).

The value can be:

MQDLV_AS_PARENT
The delivery mechanism used is based on the setting of the first parent administrative node found in the topic tree relating to this topic.

MQDLV_ALL
Non-persistent messages must be delivered to all subscribers, irrespective of durability for the MQPUT call to report success. If a delivery failure to any subscriber occurs, no other subscribers receive the message and the MQPUT fails.

MQDLV_ALL_DUR
Non-persistent messages must be delivered to all durable subscribers. Failure to deliver a non-persistent message to any non-durable subscribers does not return an error to the MQPUT call. If a delivery failure to a durable subscriber occurs, no other subscribers receive the message and the MQPUT fails.

MQDLV_ALL_AVAIL
Non-persistent messages are delivered to all subscribers that can accept the message. Failure to deliver the message to any subscriber does not prevent other subscribers from receiving the message.

PersistentMsgDelivery (MQCFIN)
The delivery mechanism for persistent messages published to this topic (parameter identifier: **MQIA_PM_DELIVERY**).

The value can be:

MQDLV_AS_PARENT
The delivery mechanism used is based on the setting of the first parent administrative node found in the topic tree relating to this topic.

MQDLV_ALL
Persistent messages must be delivered to all subscribers, irrespective of durability for the MQPUT call to report success. If a delivery failure to any subscriber occurs, no other subscribers receive the message and the MQPUT fails.

MQDLV_ALL_DUR
Persistent messages must be delivered to all durable subscribers. Failure to deliver a persistent message to any non-durable subscribers does not return an error to the MQPUT call. If a delivery failure to a durable subscriber occurs, no other subscribers receive the message and the MQPUT fails.

MQDLV_ALL_AVAIL
Persistent messages are delivered to all subscribers that can accept the message. Failure to deliver the message to any subscriber does not prevent other subscribers from receiving the message.

ProxySubscriptions (MQCFIN)
Whether a proxy subscription is to be sent for this topic, even if no local subscriptions exist, to directly connected queue managers (parameter identifier: **MQIA_PROXY_SUB**).

The value can be:

MQTA_PROXY_SUB_FORCE
A proxy subscription is sent to connected queue managers even if no local subscriptions exist.

MQTA_PROXY_SUB_FIRSTUSE
A proxy subscription is sent for this topic only when a local subscription exists.

PublicationScope (MQCFIN)
Whether this queue manager propagates publications to queue managers as part of a hierarchy or as part of a publish/subscribe cluster (parameter identifier: **MQIA_PUB_SCOPE**).

The value can be:

MQSCOPE_ALL
Publications for this topic are propagated to hierarchically connected queue managers and to publish/subscribe cluster-connected queue managers.

MQSCOPE_AS_PARENT
Whether this queue manager will propagate publications to queue managers as part of a hierarchy or as part of a publish/subscribe cluster is based on the setting of the first parent administrative node found in the topic tree relating to this topic.

This is the default value for this parameter if no value is specified.

MQSCOPE_QMGR
Publications for this topic are not propagated to other queue managers.

Note: You can override this behavior on a publication-by-publication basis, using MQPMO_SCOPE_QMGR on the Put Message Options.

QMgrName (MQCFST)
Name of local queue manager (parameter identifier: **MQCA_CLUSTER_Q_MGR_NAME**).

The maximum length of the string is MQ_Q_MGR_NAME_LENGTH.

SubscriptionScope (MQCFIN)
Whether this queue manager propagates subscriptions to queue managers as part of a hierarchy or as part of a publish/subscribe cluster (parameter identifier: **MQIA_SUB_SCOPE**).

The value can be:

MQSCOPE_ALL
Subscriptions for this topic are propagated to hierarchically connected queue managers and to publish/subscribe cluster-connected queue managers.

MQSCOPE_AS_PARENT
Whether this queue manager will propagate subscriptions to queue managers as part of a hierarchy or as part of a publish/subscribe cluster is based on the setting of the first parent administrative node found in the topic tree relating to this topic.

This is the default value for this parameter if no value is specified.

MQSCOPE_QMGR
Subscriptions for this topic are not propagated to other queue managers.

Note: You can override this behavior on a subscription-by-subscription basis, using MQSO_SCOPE_QMGR on the Subscription Descriptor or **SUBSCOPE(QMGR)** on **DEFINE SUB**.

TopicDesc (MQCFST)
Topic description (parameter identifier: **MQCA_TOPIC_DESC**).

The maximum length is MQ_TOPIC_DESC_LENGTH.

TopicName (MQCFST)
Topic object name (parameter identifier: **MQCA_TOPIC_NAME**).

The maximum length of the string is MQ_TOPIC_NAME_LENGTH.

TopicString (MQCFST)
The topic string (parameter identifier: **MQCA_TOPIC_STRING**).

The '/' character within this string has special meaning. It delimits the elements in the topic tree. A topic string can start with the '/' character but is not required to. A string starting with the '/' character is not the same as the string which starts without the '/' character. A topic string cannot end with the "/" character.

The maximum length of the string is MQ_TOPIC_STR_LENGTH.

TopicType (MQCFIN)
Whether this object is a local or cluster topic (parameter identifier: **MQIA_TOPIC_TYPE**).

The value can be:

MQTOPT_LOCAL
This object is a local topic.

MQTOPT_CLUSTER
This object is a cluster topic.

WildcardOperation (MQCFIN)
Behavior of subscriptions including wildcards made to this topic (parameter identifier: **MQIA_WILDCARD_OPERATION**).

The value can be any of the following values:

MQTA_PASSTHRU
Subscriptions made using wildcard topic names that are less specific than the topic string at this topic object will receive publications made to this topic and to topic strings more specific than this topic. This is the default supplied with IBM MQ.

MQTA_BLOCK
Subscriptions made using wildcard topic names that are less specific than the topic string at this topic object will not receive publications made to this topic or to topic strings more specific than this topic.

Authority Record attributes

Event messages relating to objects can include authority record attributes.

AuthorityRecordType (MQCFIN)
Object type (parameter identifier: **MQIACF_AUTH_REC_TYPE**).

Describes the object type whose profile is being updated, for example MQOT_Q.

AuthorizationList (MQCFIL)
Authorization list (parameter identifier: **MQIACF_AUTHORIZATION_LIST**).

The MQAUTH_* values; see Inquire Authority Records (Response).

EntityName (MQCFST)
Entity name (parameter identifier: **MQCACF_ENTITY_NAME**).

The entity name can be either a principal name or a group name.

The maximum length of the string is MQ_ENTITY_NAME_LENGTH.

EntityType (MQCFIN)
Entity type (parameter identifier: **MQIACF_ENTITY_TYPE**).

The MQZAET_* values; see Inquire Authority Records (Response).

Event message reference

Use this page to obtain an overview of information about the format of event messages.

For each instrumentation event, information is returned in both the message descriptor and message data parts of the events messages.

Related concepts
“Event message descriptions” on page 127
The event message data contains information specific to the event that was generated. This data includes the name of the queue manager and, where appropriate, the name of the queue.

Instrumentation events

Related reference
“Event message format” on page 119
Event messages are standard IBM MQ messages containing a message descriptor and message data.

“Event message MQMD (message descriptor)” on page 121
The message descriptor for an event message contains information that a system monitoring application can use, such as the message type and format, and the date and time that the message was put on the event queue.

“Event message MQCFH (PCF header)” on page 125
The message data in event messages is in programmable command format (PCF), as used in PCF command inquiries and responses. The message data consists of two parts: the event header and the event data.

Event message format

Event messages are standard IBM MQ messages containing a message descriptor and message data.

Table 9 on page 120 shows the basic structure of event messages and, in the Event data column, the names of the fields in an event message for queue service interval events.

Table 9. Event message structure for queue service interval events		
Message descriptor	**Message data**	
MQMD structure	**PCF header** **MQCFH structure**	**Event data** [1]
Structure identifier Structure version Report options Message type Expiration time Feedback code Encoding Coded character set ID Message format Message priority Persistence Message identifier Correlation identifier Backout count Reply-to queue Reply-to queue manager User identifier Accounting token Application identity data Application type Application name Put date Put time Application origin data Group identifier Message sequence number Offset Message flags Original length	Structure type Structure length Structure version Command identifier Message sequence number Control options Completion code Reason code Parameter count	Queue manager name Queue name Time since last reset Maximum number of messages on queue Number of messages put to queue Number of messages retrieved from queue

Note:

1. The parameters shown are those returned for a queue service interval event. The actual event data depends on the specific event.

In general, you need only a subset of this information for any system management programs that you write. For example, your application might need the following data:

- The name of the application causing the event
- The name of the queue manager on which the event occurred
- The queue on which the event was generated
- The event statistics

Event message MQMD (message descriptor)

The message descriptor for an event message contains information that a system monitoring application can use, such as the message type and format, and the date and time that the message was put on the event queue.

The information in the descriptor informs a system management application that the message type is MQMT_DATAGRAM, and the message format is MQFMT_EVENT.

Many of the fields in an event message contain fixed data, which is supplied by the queue manager that generated the message. The MQMD also specifies the name of the queue manager (truncated to 28 characters) that put the message.

For an event message, the MQMD structure contains the following values:

StrucId

Description: Structure identifier.

Data type: MQCHAR4.

Value: MQMD_STRUC_ID

Version

Description: Structure version number.

Data type: MQLONG.

Values:

MQMD_VERSION_1
Version-1 message descriptor structure, supported in all environments.

MQMD_VERSION_2
Version-2 message descriptor structure, supported in the following environments:

-

AIX
-

IBM i
-

Linux
-

Windows
-

z/OS

and all IBM MQ MQI clients connected to these systems.

Report

Description: Options for report messages.

Data type: MQLONG.

Value:

MQRO_NONE
No reports required.

MsgType

Description: Indicates type of message.

Data type: MQLONG.

Value: MQMT_DATAGRAM.

Expiry

Description: Message lifetime.

Data type: MQLONG.

Value:

MQEI_UNLIMITED
The message does not have an expiry time.

Feedback

Description: Feedback or reason code.

Data type: MQLONG.

Value: MQFB_NONE.

Encoding

Description: Numeric encoding of message data.

Data type: MQLONG.

Value: MQENC_NATIVE.

CodedCharSetId

Description: Character set identifier of event message data.

Data type: MQLONG.

Value: Coded character set ID (CCSID) of the queue manager generating the event.

Format

Description: Format name of message data.

Data type: MQCHAR8.

Value:

MQFMT_EVENT
Event message.

Priority

Description: Message priority.

Data type: MQLONG.

Value:

MQPRI_PRIORITY_AS_Q_DEF
The priority is that of the event queue.

Persistence

Description: Message persistence.

Data type: MQLONG.

Value:

MQPER_PERSISTENCE_AS_Q_DEF
The priority is that of the event queue.

MsgId

Description: Message identifier.

Data type: MQBYTE24.

Value: A unique value generated by the queue manager.

CorrelId

Description: Correlation identifier.

Data type: MQBYTE24.

Value: For performance, queue manager, logger, channel, bridge, and SSL events:

MQCI_NONE
No correlation identifier is specified. This is for private queues only.

For such events on a shared queue, a nonzero correlation identifier is set. This parameter is set so that you can track multiple event messages from different queue managers. The characters are specified in the following way:

1-4 Product identifier ('CSQ ')

5-8 Queue sharing group name

9 Queue manager identifier

10-17 Time stamp

18-24 Nulls

For configuration and command events:

A unique nonzero correlation identifier
All messages relating to the same event have the same CorrelId.

BackoutCount

Description: Backout counter.

Data type: MQLONG.

Value: 0.

ReplyToQ

Description: Name of reply queue.

Data type: MQCHAR48.

Values: Blank.

ReplyToQMgr

Description: Name of reply queue manager.

Data type: MQCHAR48.

Value: The queue manager name at the originating system.

UserIdentifier

Description: Identifies the application that originated the message.

Data type: MQCHAR12.

Value: Blank.

AccountingToken

Description: Accounting token that allows an application to charge for work done as a result of the message.

Data type: MQBYTE32.

Value: MQACT_NONE.

ApplIdentityData

Description: Application data relating to identity.

Data type: MQCHAR32.

Values: Blank.

PutApplType

Description: Type of application that put the message.

Data type: MQLONG.

Value:

MQAT_QMGR
Queue manager generated message.

PutApplName

Description: Name of application that put the message.

Data type: MQCHAR28.

Value: The queue manager name at the originating system.

PutDate

Description: Date when message was put.

Data type: MQCHAR8.

Value: As generated by the queue manager.

PutTime

Description: Time when message was put.

Data type: MQCHAR8.

Value: As generated by the queue manager.

ApplOriginData

Description: Application data relating to origin.

Data type: MQCHAR4.

Value: Blank.

Note: If *Version* is MQMD_VERSION_2, the following additional fields are present:

GroupId

Description: Identifies to which message group or logical message the physical message belongs.

Data type: MQBYTE24.

Value: **MQGI_NONE**
No group identifier specified.

MsgSeqNumber

Description: Sequence number of logical message within group.

Data type: MQLONG.

Value: 1.

Offset

Description: Offset of data in physical message from start of logical message.

Data type: MQLONG.

Value: 0.

MsgFlags

Description: Message flags that specify attributes of the message or control its processing.

Data type: MQLONG.

Value: MQMF_NONE.

OriginalLength

Description: Length of original message.

Data type: MQLONG.

Value: MQOL_UNDEFINED.

Event message MQCFH (PCF header)

The message data in event messages is in programmable command format (PCF), as used in PCF command inquiries and responses. The message data consists of two parts: the event header and the event data.

The MQCFH header specifies the following information:

- The category of event: whether the event is a queue manager, performance, channel, configuration, command, or logger event.
- A reason code specifying the cause of the event. For events caused by MQI calls, this reason code is the same as the reason code for the MQI call.

Reason codes have names that begin with the characters MQRC_. For example, the reason code MQRC_PUT_INHIBITED is generated when an application attempts to put a message on a queue that is not enabled for puts.

For an event, the MQCFH structure contains the following values:

Type

Description: Structure type that identifies the content of the message.

Data type: MQLONG.

Value: **MQCFT_EVENT**
Message is reporting an event.

StrucLength

Description: Structure length.

Data type: MQLONG.

Value:

MQCFH_STRUC_LENGTH
Length in bytes of MQCFH structure.

Version

Description: Structure version number.

Data type: MQLONG.

Values:

MQCFH_VERSION_1
Version-1 in all events except configuration and command events.

MQCFH_VERSION_2
Version-2 for configuration events.

MQCFH_VERSION_3
Version-3 for command events.

Command

Description: Command identifier. This identifies the event category.

Data type: MQLONG.

Values:

MQCMD_Q_MGR_EVENT
Queue manager event.

MQCMD_PERFM_EVENT
Performance event.

MQCMD_CHANNEL_EVENT
Channel event.

MQCMD_CONFIG_EVENT
Configuration event.

MQCMD_COMMAND_EVENT
Command event.

MQCMD_LOGGER_EVENT
Logger event.

MsgSeqNumber

Description: Message sequence number. This is the sequence number of the message within a group of related messages.

Data type: MQLONG.

Values:

1
For change object configuration events with attribute values before the changes, and for all other types of events.

2
For change object configuration events with the attribute values after the changes

Control

Description: Control options.

Data type: MQLONG.

Values:

MQCFC_LAST
For change object configuration events with attribute values after the changes, and for all other types of events.

MQCFC_NOT_LAST
For Change Object configurations events only, with the attribute values from before the changes.

CompCode

Description: Completion code.

Data type: MQLONG.

Values:

MQCC_OK
Event reporting OK condition.

MQCC_WARNING
Event reporting warning condition. All events have this completion code, unless otherwise specified.

Reason

Description: Reason code qualifying completion code.

Data type: MQLONG.

Values: MQRC_* Dependent on the event being reported.

Note: Events with the same reason code are further identified by the **ReasonQualifier** parameter in the event data.

ParameterCount

Description: Count of parameter structures. This is the number of parameter structures that follow the MQCFH structure. A group structure (MQCFGR), and its included parameter structures, are counted as one structure only.

Data type: MQLONG.

Values: 0 or greater.

Event message descriptions

The event message data contains information specific to the event that was generated. This data includes the name of the queue manager and, where appropriate, the name of the queue.

The data structures returned depend on which particular event was generated. In addition, for some events, certain parameters of the structures are optional, and are returned only if they contain information that is relevant to the circumstances giving rise to the event. The values in the data structures depend on the circumstances that caused the event to be generated.

Note:

1. The PCF structures in the message data are not returned in a defined order. They must be identified from the parameter identifiers shown in the description.
2. Events are available on all platforms, unless specific limitations are shown at the start of an event description.

Alias Base Queue Type Error

Event name:	Alias Base Queue Type Error.
Reason code in MQCFH:	"2001 (07D1) (RC2001): MQRC_ALIAS_BASE_Q_TYPE_ERROR" on page 1153. Alias base queue not a valid type.
Event description:	An MQOPEN or MQPUT1 call was issued specifying an alias queue as the destination, but the *BaseObjectName* in the alias queue definition resolves to a queue that is not a local queue, or local definition of a remote queue.
Event type:	Local.
Platforms:	All.
Event queue:	SYSTEM.ADMIN.QMGR.EVENT.

Event data

QMgrName

Description: Name of the queue manager generating the event.

Identifier: MQCA_Q_MGR_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: Always.

QName

Description: Queue name from object descriptor (MQOD).

Identifier: MQCA_Q_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_NAME_LENGTH.

Returned: Always.

BaseObjectName

Description: Object name to which the alias resolves.

Identifier: MQCA_BASE_OBJECT_NAME. For compatibility with existing applications you can still use MQCA_BASE_Q_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_NAME_LENGTH.

Returned: Always.

QType

Description: Type of queue to which the alias resolves.

Identifier: MQIA_Q_TYPE.

Data type: MQCFIN.

Values:

MQQT_ALIAS
Alias queue definition.

MQQT_MODEL
Model queue definition.

Returned: Always.

ApplType

Description: Type of the application making the call that caused the event.

Identifier: MQIA_APPL_TYPE.

Data type: MQCFIN.

Returned: Always.

ApplName

Description: Name of the application making the call that caused the event.

Identifier: MQCACF_APPL_NAME.

Data type: MQCFST.

Maximum length: MQ_APPL_NAME_LENGTH.

Returned: Always.

ObjectQMgrName

Description: Name of the object queue manager.

Identifier: MQCACF_OBJECT_Q_MGR_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: If the *ObjectName* in the object descriptor (MQOD), when the object was opened, is not the queue manager currently connected.

ConnName

Description: Connection name for client connection.

Identifier: MQCACH_CONNECTION_NAME.

Data type: MQCFST.

Maximum length: MQ_CONN_NAME_LENGTH.

Returned: If the application making the MQI call that caused the event is a client attached application.

ChannelName

Description: Channel name for client connection.

Identifier: MQCACH_CHANNEL_NAME.

Data type: MQCFST.

Maximum length: MQ_CHANNEL_NAME_LENGTH

Returned: If the application making the MQI call that caused the event is a client attached application.

Multi *Authorized (type 1)*

Event name:	Authorized (type 1).
Reason code in MQCFH:	3391 (0D3F) (RC3391): MQRCCF_AUTHORIZED
Event description:	On an MQCONN or system connection call, the user is authorized to connect to the queue manager. **ReasonQualifier** identifies the nature of the event.
Event type:	Authority.
Platforms:	Multiplatforms only.
Event queue:	SYSTEM.ADMIN.QMGR.EVENT.

Event data

QMgrName

Description: Name of the queue manager generating the event.

Identifier: MQCA_Q_MGR_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: Always.

ReasonQualifier

Description: Identifier for type 1 authority extended events.

Identifier: MQIACF_REASON_QUALIFIER.

Data type: MQCFIN.

Values:

MQRQ_CONN_AUTHORIZED
Connection authorized.

Returned: Always.

UserIdentifier

Description: User identifier that caused the authorization check.

Identifier: MQCACF_USER_IDENTIFIER.

Data type: MQCFST.

Maximum length: MQ_USER_ID_LENGTH.

Returned: Always.

ApplType

Description: Type of application causing the event.

Identifier: MQIA_APPL_TYPE.

Data type: MQCFIN.

Returned: Always.

ApplName

Description: Name of the application causing the event.

Identifier: MQCACF_APPL_NAME.

Data type: MQCFST.

Maximum length: MQ_APPL_NAME_LENGTH.

Returned: Always.

ConnName

Description: Connection name for client connection.

Identifier: MQCACH_CONNECTION_NAME.

Data type: MQCFST.

Maximum length: MQ_CONN_NAME_LENGTH.

Returned: If the application making the MQI call that caused the event is a client attached application.

ChannelName

Description: Channel name for client connection.

Identifier: MQCACH_CHANNEL_NAME.

Data type: MQCFST.

Maximum length: MQ_CHANNEL_NAME_LENGTH.

Returned: If the application making the MQI call that caused the event is a client attached application.

CSPUserIdentifier

Description: The user identifier from the Connection Security Parameters (MQCSP) structure. If an authentication token is specified in the MQCSP structure, the value of this parameter is *TOKEN.

Identifier: MQCACF_CSP_USER_IDENTIFIER.

Data type: MQCFST.

Maximum length: MQ_CLIENT_USER_ID_LENGTH.

Returned: If the application making the MQI call that caused the event supplied a user ID or an authentication token in the MQCSP structure.

ConnectionIdentifier

Description: The connection identifier associated with successful connections.

Identifier: MQBACF_CONNECTION_ID.

Data type: MQCFBS.

Maximum length: MQ_CONNECTION_ID_LENGTH.

Returned: Always.

ConnectionTag

Description: The connection tag associated the application.

Identifier: MQBACF_CONN_TAG.

Data type: MQCFBS.

Maximum length: MQ_CONN_TAG_LENGTH.

Returned: Always.

TLS Security Protocol

Description: The TLS Protocol version used to secure the client connection.

Identifier: MQIACH_SECURITY_PROTOCOL

Data type: MQCFIN

Returned: If the connection is secured via TLS.

TLS CipherSpec

Description: The TLS CipherSpec used to secure the client connection.

Identifier: MQCACH_SSL_CIPHER_SPEC.

Data type: MQCFST.

Maximum length: MQ_SSL_CIPHER_SPEC_LENGTH.

Returned: If the connection is secured via TLS.

SSL Peer name

Description: The SSL Peer name of a certificate used by the client application.

Identifier: MQCACH_SSL_PEER_NAME.

Data type: MQCFST.

Maximum length: MQ_SSL_SHORT_PEER_NAME_LENGTH.

Returned: If the connection is secured via TLS and the client application provided a TLS certificate.

SSL Issuer name

Description: The SSL issuer name of a certificate used by the client application.

Identifier: MQCA_SSL_CERT_ISSUER_NAME

Data type: MQCFST

Maximum length: MQ_SSL_SHORT_PEER_NAME_LENGTH

Returned: If the connection is secured via TLS and the client application provided a TLS certificate.

Remote Product

Description: The remote product identifier.

Identifier: MQCACH_REMOTE_PRODUCT.

Data type: MQCFST.

Maximum length: MQ_REMOTE_PRODUCT_LENGTH.

Returned: Always.

Remote Version

Description: The remote product version.

Identifier: MQCACH_REMOTE_VERSION.

Data type: MQCFST.

Maximum length: MQ_REMOTE_VERSION_LENGTH.

Returned: Always.

SSL Quantum Safe Algorithm

Description: The quantum safe algorithm used to secure the connection.

Identifier: MQIACH_SSL_QS.

Data type: MQCFIN.

Returned: If the connection is secured by TLS and a quantum safe algorithm was used.

Multi *Authorized (type 2)*

Event name:	Authorized (type 2)
Reason code in MQCFH:	3391 (0D3F) (RC3391): MQRCCF_AUTHORIZED
Event description:	On an MQOPEN or MQPUT1 call, the user is authorized to open the object for the options specified.
Event type:	Authority
Platforms:	Multiplatforms only
Event queue:	SYSTEM.ADMIN.QMGR.EVENT

Event data

QMgrName

Description: Name of the queue manager generating the event.

Identifier: MQCA_Q_MGR_NAME

Data type: MQCFST

Maximum length: MQ_Q_MGR_NAME_LENGTH

Returned: Always

ReasonQualifier

Description: Identifier for type 2 authority extended events.

Identifier: MQIACF_REASON_QUALIFIER

Data type: MQCFIN

Values:

MQRQ_OPEN_AUTHORIZED
Open authorized

Returned: Always

Options

Description: Options specified on the MQOPEN call.

Identifier: MQIACF_OPEN_OPTIONS

Data type: MQCFIN

Returned: Always

UserIdentifier

Description: User identifier that caused the authorization check.

Identifier: MQCACF_USER_IDENTIFIER

Data type: MQCFST

Maximum length: MQ_USER_ID_LENGTH

Returned: Always

ApplType

Description: Type of application causing the event.

Identifier: MQIA_APPL_TYPE

Data type: MQCFIN

Returned: Always

ApplName

Description: Name of the application causing the event.

Identifier: MQCACF_APPL_NAME

Data type: MQCFST

Maximum length: MQ_APPL_NAME_LENGTH

Returned: Always

ObjectQMgrName

Description: Object queue manager name from object descriptor (MQOD).

Identifier: MQCACF_OBJECT_Q_MGR_NAME

Data type: MQCFST

Maximum length: MQ_Q_MGR_NAME_LENGTH

Returned: If the **ObjectQMgrName** in the object descriptor (MQOD) when the object was opened is not the queue manager currently connected.

QName

Description: Object name from object descriptor (MQOD).

Identifier: MQCA_Q_NAME

Data type: MQCFST

Maximum length: MQ_Q_NAME_LENGTH

Returned: If the object opened is a queue object.

ProcessName

Description: Name of process object from object descriptor (MQOD).

Identifier: MQCA_PROCESS_NAME

Data type: MQCFST

Maximum length: MQ_PROCESS_NAME_LENGTH

Returned: If the object opened is a process object.

TopicString

Description: Topic string being subscribed to, or opened.

Identifier: MQCA_TOPIC_STRING

Data type: MQCFST

Maximum length: MQ_TOPIC_STR_LENGTH

Returned: If the object opened is a topic object.

AdminTopicNames

Description: List of topic admin objects against which authority is checked.

Identifier: MQCA_ADMIN_TOPIC_NAME

Data type: MQCFSL

Maximum length: MQ_TOPIC_NAME_LENGTH

Returned: If the object opened is a topic object.

ObjectType

Description: Object type from object descriptor (MQOD).

Identifier: MQIACF_OBJECT_TYPE

Data type: MQCFIN

Values:

MQOT_NAMELIST
Namelist

MQOT_PROCESS
Process

MQOT_Q
Queue

MQOT_Q_MGR
Queue manager

MQOT_TOPIC
Topic

Returned: Always

NamelistName

Description: Object name from object descriptor (MQOD).

Identifier: MQCA_NAMELIST_NAME

Data type: MQCFST

Maximum length: MQ_NAMELIST_NAME_LENGTH

Returned: If the object opened is a namelist object.

ConnName

Description: Connection name for client connection.

Identifier: MQCACH_CONNECTION_NAME

Data type: MQCFST

Maximum length: MQ_CONN_NAME_LENGTH

Returned: If the application making the MQI call that caused the event is a client attached application.

ChannelName

Description: Channel name for client connection.

Identifier: MQCACH_CHANNEL_NAME

Data type: MQCFST

Maximum length: MQ_CHANNEL_NAME_LENGTH

Returned: If the application making the MQI call that caused the event is a client attached application.

ConnectionIdentifier

Description: The connection identifier associated with successful connections.

Identifier: MQBACF_CONNECTION_ID

Data type: MQCFBS

Maximum length: MQ_CONNECTION_ID_LENGTH

Returned: Always.

ConnectionTag

Description: The connection tag associated the application.

Identifier: MQBACF_CONN_TAG

Data type: MQCFBS

Maximum length: MQ_CONN_TAG_LENGTH

Returned: Always.

EventDuplicateCount

Description: How many duplicate copies of this message have been processed. Range can be determined by examining the **MQMD.PutTime**, **MQMD.PutDate** and **EventDuplicateFrom** values.

Identifier: MQIACF_EVENT_DUPLICATE_COUNT

Data type: MQCFIN

Returned: If there have been duplicate copies cached.

EventDuplicateFrom

Description: The time of first duplicate event, in ISO 8601 format.

Identifier: MQCACF_EVENT_DUPLICATE_FROM

Data type: MQCFST

Maximum length: MQ_ISOTIME_LENGTH

Returned: If there have been duplicate copies cached.

Multi *Authorized (type 3)*

Event name:	Authorized (type 3).
Reason code in MQCFH:	3391 (0D3F) (RC3391): MQRCCF_AUTHORIZED
Event description:	On an MQSUB call, the user is authorized to subscribe to the specified topic.
Event type:	Authority.
Platforms:	Multiplatforms only.
Event queue:	SYSTEM.ADMIN.QMGR.EVENT.

Event data

QMgrName

Description: Name of the queue manager generating the event.
Identifier: MQCA_Q_MGR_NAME.
Data type: MQCFST.
Maximum length: MQ_Q_MGR_NAME_LENGTH.
Returned: Always.

ReasonQualifier

Description: Identifier for type 3 authority extended events.
Identifier: MQIACF_REASON_QUALIFIER.
Data type: MQCFIN.
Values:
MQRQ_SUB_AUTHORIZED
Subscribe authorized.
Returned: Always.

Options

Description: Options specified on the MQSUB call.
Identifier: MQIACF_SUB_OPTIONS.
Data type: MQCFIN.
Returned: Always.

UserIdentifier

Description: User identifier that caused the authorization check.
Identifier: MQCACF_USER_IDENTIFIER.
Data type: MQCFST.
Maximum length: MQ_USER_ID_LENGTH.
Returned: Always.

ApplType

Description: Type of application causing the event.
Identifier: MQIA_APPL_TYPE.

Data type: MQCFIN.

Returned: Always.

ApplName

Description: Name of the application causing the event.

Identifier: MQCACF_APPL_NAME.

Data type: MQCFST.

Maximum length: MQ_APPL_NAME_LENGTH.

Returned: Always.

TopicString

Description: Topic string being subscribed to or opened.

Identifier: MQCA_TOPIC_STRING.

Data type: MQCFST.

Maximum length: MQ_TOPIC_STR_LENGTH.

Returned: If the object opened is a topic object.

AdminTopicNames

Description: List of topic admin objects against which authority is checked.

Identifier: MQCA_ADMIN_TOPIC_NAME.

Data type: MQCFSL.

Maximum length: MQ_TOPIC_NAME_LENGTH.

Returned: If the object opened is a topic object.

ConnName

Description: Connection name for client connection.

Identifier: MQCACH_CONNECTION_NAME.

Data type: MQCFST.

Maximum length: MQ_CONN_NAME_LENGTH.

Returned: If the application making the MQI call that caused the event is a client attached application.

ChannelName

Description: Channel name for client connection.

Identifier: MQCACH_CHANNEL_NAME.

Data type: MQCFST.

Maximum length: MQ_CHANNEL_NAME_LENGTH.

Returned: If the application making the MQI call that caused the event is a client attached application.

ConnectionIdentifier

Description: The connection identifier associated with successful connections.

Identifier: MQBACF_CONNECTION_ID.

Data type: MQCFBS.

Maximum length: MQ_CONNECTION_ID_LENGTH.

Returned: Always.

ConnectionTag

Description: The connection tag associated the application

Identifier: MQBACF_CONN_TAG.

Data type: MQCFBS.

Maximum length: MQ_CONN_TAG_LENGTH.

Returned: Always.

Multi *Authorized (type 4)*

Event name:	Authorized (type 4).
Reason code in MQCFH:	3391 (0D3F) (RC3391): MQRCCF_AUTHORIZED
Event description:	On an MQSUB call, the user is authorized to use the destination queue with the required level of access. This event is only returned for subscriptions using non-managed destination queues.
Event type:	Authority.
Platforms:	Multiplatforms only.
Event queue:	SYSTEM.ADMIN.QMGR.EVENT.

Event data

QMgrName

Description: Name of the queue manager generating the event.

Identifier: MQCA_Q_MGR_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: Always.

ReasonQualifier

Description: Identifier for type 4 authority extended events.

Identifier: MQIACF_REASON_QUALIFIER.

Data type: MQCFIN.

Values: **MQRQ_SUB_DEST_AUTHORIZED**
Subscription destination queue usage authorized.

Returned: Always.

Options

Description: Options specified on the MQSUB call..

Identifier: MQIACF_SUB_OPTIONS.

Data type: MQCFIN.

Returned: Always.

UserIdentifier

Description: User identifier that caused the authorization check.

Identifier: MQCACF_USER_IDENTIFIER.

Data type: MQCFST.

Maximum length: MQ_USER_ID_LENGTH.

Returned: Always.

ApplType

Description: Type of application causing the event.

Identifier: MQIA_APPL_TYPE.

Data type: MQCFIN.

Returned: Always.

ApplName

Description: Name of the application causing the event.

Identifier: MQCACF_APPL_NAME.

Data type: MQCFST.

Maximum length: MQ_APPL_NAME_LENGTH.

Returned: Always.

TopicString

Description: Topic string being subscribed to.

Identifier: MQCA_TOPIC_STRING.

Data type: MQCFST.

Maximum length: MQ_TOPIC_STR_LENGTH.

Returned: If the object opened is a topic object.

DestQMgrName

Description: Hosting queue manager name of the subscription's destination queue.

Identifier: MQCACF_OBJECT_Q_MGR_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: If the queue manager hosting the destination queue is not the queue manager to which the application is currently connected.

DestQName

Description: The name of the destination queue of the subscription.

Identifier: MQCA_Q_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_NAME_LENGTH..

Returned: Always.

DestOpenOptions

Description: The open options requested for the destination queue.

Identifier: MQIACF_OPEN_OPTIONS.

Data type: MQCFIN.

Returned: Always.

ConnName

Description: Connection name for client connection.

Identifier: MQCACH_CONNECTION_NAME.

Data type: MQCFST.

Maximum length: MQ_CONN_NAME_LENGTH.

Returned: If the application making the MQI call that caused the event is a client attached application.

ChannelName

Description: Channel name for client connection.

Identifier: MQCACH_CHANNEL_NAME.

Data type: MQCFST.

Maximum length: MQ_CHANNEL_NAME_LENGTH.

Returned: If the application making the MQI call that caused the event is a client attached application.

ConnectionIdentifier

Description: The connection identifier associated with successful connections.

Identifier: MQBACF_CONNECTION_ID.

Data type: MQCFBS.

Maximum length: MQ_CONNECTION_ID_LENGTH.

Returned: Always.

ConnectionTag

Description: The connection tag associated the application

Identifier: MQBACF_CONN_TAG.

Data type: MQCFBS.

Maximum length: MQ_CONN_TAG_LENGTH.

Returned: Always.

Bridge Started

Event name:	Bridge Started.
Reason code in MQCFH:	"2125 (084D) (RC2125): MQRC_BRIDGE_STARTED" on page 1207. Bridge started.
Event description:	The IMS bridge has been started.
Event type:	IMS bridge.
Platforms:	IBM MQ for z/OS only.
Event queue:	SYSTEM.ADMIN.CHANNEL.EVENT.

Event data

QMgrName

Description: Name of the queue manager generating the event.

Identifier: MQCA_Q_MGR_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: Always.

BridgeType

Description: Bridge type.

Identifier: MQIACF_BRIDGE_TYPE.

Data type: MQCFIN.

Values:

MQBT_OTMA
OTMA bridge.

Returned: Always.

BridgeName

Description: Bridge name. For bridges of type MQBT_OTMA, the name is of the form `XCFgroupXCFmember`, where `XCFgroup` is the XCF group name to which both IMS and IBM MQ belong. `XCFmember` is the XCF member name of the IMS system.

Identifier: MQCACF_BRIDGE_NAME.

Data type: MQCFST.

Maximum length: MQ_BRIDGE_NAME_LENGTH.

Returned: Always.

Bridge Stopped

Event name:	Bridge Stopped.
Reason code in MQCFH:	"2126 (084E) (RC2126): MQRC_BRIDGE_STOPPED" on page 1207. Bridge stopped.
Event description:	The IMS bridge has been stopped.

Event type:	IMS bridge.
Platforms:	IBM MQ for z/OS only.
Event queue:	SYSTEM.ADMIN.CHANNEL.EVENT.

Event data

QMgrName

Description: Name of the queue manager generating the event.

Identifier: MQCA_Q_MGR_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: Always.

ReasonQualifier

Description: Identifier that qualifies the reason code in MQCFH.

Identifier: MQIACF_REASON_QUALIFIER.

Data type: MQCFIN.

Values:

MQRQ_BRIDGE_STOPPED_OK
Bridge has been stopped with either a zero return code or a warning return code. For MQBT_OTMA bridges, one side or the other issued a normal IXCLEAVE request.

MQRQ_BRIDGE_STOPPED_ERROR
Bridge has been stopped but there is an error reported.

Returned: Always.

BridgeType

Description: Bridge type.

Identifier: MQIACF_BRIDGE_TYPE.

Data type: MQCFIN.

Value:

MQBT_OTMA
OTMA bridge.

Returned: Always.

BridgeName

Description: Bridge name. For bridges of type MQBT_OTMA, the name is of the form XCF`group`XCF`member`, where XCF`group` is the XCF group name to which both IMS and IBM MQ belong. XCF`member` is the XCF member name of the IMS system.

Identifier: MQCACF_BRIDGE_NAME.

Data type: MQCFST.

Maximum length: MQ_BRIDGE_NAME_LENGTH.

Returned: Always.

ErrorIdentifier

Description: When a bridge is stopped because of an error, this code identifies the error. If the event reports a bridge stop failure, the IMS sense code is set.

Identifier: MQIACF_ERROR_IDENTIFIER.

Data type: MQCFIN.

Returned: If *ReasonQualifier* is MQRQ_BRIDGE_STOPPED_ERROR.

Change Authority Record

Event name:	Change Authority Record
Reason code in MQCFH:	"2368 (0940) (RC2368): MQRC_CONFIG_CHANGE_OBJECT" on page 1286. Object changed.
Event description:	A Set Authority Record command was issued that successfully changed an existing authority record.
Event type:	Configuration
Platforms:	All except z/OS.
Event queue:	SYSTEM.ADMIN.CONFIG.EVENT.

Note that two event messages are generated for the change authority record event. The first has the authority record attribute values *before* the change; the second has the attribute values *after* the change.

Event data

EventQMgr

Description: The queue manager where the command or call was entered. That is, the queue manager where the command is processed and that generates the event is in the MQMD of the event message.

Identifier: MQCACF_EVENT_Q_MGR

Data type: MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: Always.

EventUserId

Description: The user ID that issued the command or call that generated the event.

This is the same user ID that is used to check the authority to issue the command or call; for commands received from a queue, this is also the user identifier (**UserIdentifier**) from the message descriptor of the command message.

.

Identifier: MQCACF_EVENT_USER_ID

Data type: MQCFST.

Maximum length: MQ_USER_ID_LENGTH.

Returned: Always.

EventOrigin

Description: The origin of the action causing the event.

Identifier: MQIACF_EVENT_ORIGIN

Data type: MQCFIN.

Values:

MQEVO_CONSOLE
Console command (runmqsc or setmqaut)

MQEVO_INTERNAL
Directly by queue manager

MQEVO_MSG
Command message on SYSTEM.ADMIN.COMMAND.QUEUE

Returned: Always

EventAccountingToken

Description: For commands received as a message (MQEVO_MSG), the accounting token (**AccountingToken**) from the message descriptor of the command message.

Identifier: MQBACF_EVENT_ACCOUNTING_TOKEN

Data type: MQCFBS

Maximum length: MQ_ACCOUNTING_TOKEN_LENGTH

Returned: Only if **EventOrigin** is MQEVO_MSG.

EventApplIdentity

Description: For commands received as a message (MQEVO_MSG), application identity data (**ApplIdentityData**) from the message descriptor of the command message.

Identifier: MQMQCACF_EVENT_APPL_IDENTITY

Data type: MQCFST

Maximum length: MQ_APPL_IDENTITY_DATA_LENGTH

Returned: Only if **EventOrigin** is MQEVO_MSG.

EventApplType

Description: For commands received as a message (MQEVO_MSG), the type of application (**PutApplType**) from the message descriptor of the command message.

Identifier: MQIACF_EVENT_APPL_TYPE

Data type: MQCFIN

Values:

Returned: Only if **EventOrigin** is MQEVO_MSG.

EventApplName

Description: For commands received as a message (MQEVO_MSG), the name of the application (**PutApplName**) from the message descriptor of the command message.

Identifier: MQCACF_EVENT_APPL_NAME

Data type: MQCFST

Maximum length: MQ_APPL_NAME_LENGTH

Returned: Only if **EventOrigin** is MQEVO_MSG.

EventApplOrigin

Description: For commands received as a message (MQEVO_MSG), the application origin data (**ApplOriginData**) from the message descriptor of the command message.

Identifier: MQCACF_EVENT_APPL_ORIGIN

Data type: MQCFST

Maximum length: MQ_APPL_ORIGIN_DATA_LENGTH

Returned: Only if **EventOrigin** is MQEVO_MSG.

ObjectType

Description: Object type

Identifier: MQIACF_OBJECT_TYPE

Data type: MQCFIN

Values: MQOT_AUTH_REC

Returned: Always

ProfileName

Description: Object or generic profile name

Identifier: MQCACF_AUTH_PROFILE_NAME

Data type: MQCFST

Maximum length: MQ_AUTH_PROFILE_NAME_LENGTH

Returned: Always

Object attributes

A parameter structure is returned for each attribute of the object. The attributes returned depend on the object type. For more information see "Authority Record attributes" on page 119.

Change object

Event name:	Change object.
Reason code in MQCFH:	"2368 (0940) (RC2368): MQRC_CONFIG_CHANGE_OBJECT" on page 1286. Existing object changed.
Event description:	An ALTER or DEFINE REPLACE command or an MQSET call was issued that successfully changed an existing object.
Event type:	Configuration.
Platforms:	All.
Event queue:	SYSTEM.ADMIN.CONFIG.EVENT.

Note: Two event messages are generated for the change object event. The first has the object attribute values **before** the change, the second has the attribute values **after** the change.

Event data

EventUserId

Description: The user ID that issued the command or call that generated the event. (This is the same user ID that is used to check the authority to issue the command or call; for commands received from a queue, this is also the user identifier (UserIdentifier) from the MQMD of the command message).

Identifier: MQCACF_EVENT_USER_ID.

Datatype: MQCFST.

Maximum length: MQ_USER_ID_LENGTH.

Returned: Always.

EventOrigin

Description: The origin of the action causing the event.

Identifier: MQIACF_EVENT_ORIGIN.

Datatype: MQCFIN.

Values:

MQEVO_CONSOLE
Console command.

MQEVO_INIT
Initialization input data set command.

MQEVO_INTERNAL
Directly by queue manager.

MQEVO_MQSET
MQSET call.

MQEVO_MSG
Command message on SYSTEM.COMMAND.INPUT.

MQEVO_OTHER
None of the above.

Returned: Always.

EventQMgr

Description: The queue manager where the command or call was entered. (The queue manager where the command is executed and that generates the event is in the MQMD of the event message).

Identifier: MQCACF_EVENT_Q_MGR.

Datatype: MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: Always.

EventAccountingToken

Description: For commands received as a message (MQEVO_MSG), the accounting token (AccountingToken) from the MQMD of the command message.

Identifier: MQBACF_EVENT_ACCOUNTING_TOKEN.

Datatype: MQCFBS.

Maximum length: MQ_ACCOUNTING_TOKEN_LENGTH.

Returned: Only if EventOrigin is MQEVO_MSG.

EventApplIdentity

Description: For commands received as a message (MQEVO_MSG), application identity data (ApplIdentityData) from the MQMD of the command message.

Identifier: MQCACF_EVENT_APPL_IDENTITY.

Datatype: MQCFST.

Maximum length: MQ_APPL_IDENTITY_DATA_LENGTH.

Returned: Only if EventOrigin is MQEVO_MSG.

EventApplType

Description: For commands received as a message (MQEVO_MSG), the type of application (PutApplType) from the MQMD of the command message.

Identifier: MQIACF_EVENT_APPL_TYPE.

Datatype: MQCFIN.

Returned: Only if EventOrigin is MQEVO_MSG.

EventApplName

Description: For commands received as a message (MQEVO_MSG), the name of the application (PutApplName) from the MQMD of the command message.

Identifier: MQCACF_EVENT_APPL_NAME.

Datatype: MQCFST.

Maximum length: MQ_APPL_NAME_LENGTH.

Returned: Only if EventOrigin is MQEVO_MSG.

EventApplOrigin

Description: For commands received as a message (MQEVO_MSG), the application origin data (ApplOriginData) from the MQMD of the command message.

Identifier: MQCACF_EVENT_APPL_ORIGIN.

Datatype: MQCFST.

Maximum length: MQ_APPL_ORIGIN_DATA_LENGTH.

Returned: Only if EventOrigin is MQEVO_MSG.

ObjectType

Description: Object type:

Identifier: MQIACF_OBJECT_TYPE.

Datatype: MQCFIN.

Values:

MQOT_CHANNEL
Channel.

MQOT_CHLAUTH
Channel authentication record.

MQOT_NAMELIST
Namelist.

MQOT_NONE
No object.

MQOT_PROCESS
Process.

MQOT_Q
Queue.

MQOT_Q_MGR
Queue manager.

MQOT_STORAGE_CLASS
Storage class.

MQOT_AUTH_INFO
Authentication information.

MQOT_CF_STRUC
CF structure.

MQOT_TOPIC
Topic.

MQOT_COMM_INFO
Communication information.

MQOT_LISTENER
Channel Listener.

Returned: Always.

ObjectName

Description: Object name:

Identifier : Identifier will be according to object type.

- MQCACH_CHANNEL_NAME
- MQCA_NAMELIST_NAME
- MQCA_PROCESS_NAME
- MQCA_Q_NAME
- MQCA_Q_MGR_NAME
- MQCA_STORAGE_CLASS
- MQCA_AUTH_INFO_NAME
- MQCA_CF_STRUC_NAME
- MQCA_TOPIC_NAME
- MQCA_COMM_INFO_NAME
- MQCACH_LISTENER_NAME

Note: MQCACH_CHANNEL_NAME can also be used for channel authentication.

Datatype: MQCFST.

Maximum length: MQ_OBJECT_NAME_LENGTH.

Returned: Always

Disposition

Description: Object disposition:

Identifier: MQIA_QSG_DISP.

Datatype: MQCFIN.

Values:

MQQSGD_Q_MGR
Object resides on page set of queue manager.

MQQSGD_SHARED
Object resides in shared repository and messages are shared in coupling facility.

MQQSGD_GROUP
Object resides in shared repository.

MQQSGD_COPY
Object resides on page set of queue manager and is a local copy of a GROUP object.

Returned: Always, except for queue manager and CF structure objects.

Object attributes

A parameter structure is returned for each attribute of the object. The attributes returned depend on the object type. For more information see "Object attributes for event data" on page 74.

Channel Activated

Event name:	Channel Activated.
Reason code in MQCFH:	"2295 (08F7) (RC2295): MQRC_CHANNEL_ACTIVATED" on page 1259. Channel activated.
Event description:	This condition is detected when a channel that has been waiting to become active, and for which a Channel Not Activated event has been generated, is now able to become active, because an active slot has been released by another channel. This event is not generated for a channel that is able to become active without waiting for an active slot to be released.
Event type:	Channel.
Platforms:	All.
Event queue:	SYSTEM.ADMIN.CHANNEL.EVENT.

Event data

QMgrName

Description: Name of the queue manager generating the event.

Identifier: MQCA_Q_MGR_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: Always.

ChannelName

Description: Channel Name.

Identifier: MQCACH_CHANNEL_NAME.

Data type: MQCFST.

Maximum length: MQ_CHANNEL_NAME_LENGTH.

Returned: Always.

XmitQName

Description: Transmission queue name.

Identifier: MQCACH_XMIT_Q_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_NAME_LENGTH.

Returned: For sender, server, cluster-sender, and cluster-receiver channels only.

ConnectionName

Description: If the channel has successfully established a TCP connection, this is the Internet address. Otherwise it is the contents of the *ConnectionName* field in the channel definition.

Identifier: MQCACH_CONNECTION_NAME.

Data type: MQCFST.

Maximum length: MQ_CONN_NAME_LENGTH.

Returned: Only for commands that do not contain a generic name.

Channel Auto-definition Error

Event name:	Channel Auto-definition Error.
Reason code in MQCFH:	"2234 (08BA) (RC2234): MQRC_CHANNEL_AUTO_DEF_ERROR" on page 1239. Automatic channel definition failed.
Event description:	This condition is detected when the automatic definition of a channel fails; this may be because an error occurred during the definition process, or because the channel automatic-definition exit inhibited the definition. Additional information indicating the reason for the failure is returned in the event message.
Event type:	Channel.
Platforms:	All, except IBM MQ for z/OS.
Event queue:	SYSTEM.ADMIN.CHANNEL.EVENT.

Event data

QMgrName

Description: Name of the queue manager generating the event.

Identifier: MQCA_Q_MGR_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: Always.

ChannelName

Description: Name of the channel for which the auto-definiton has failed.

Identifier: MQCACH_CHANNEL_NAME.

Data type: MQCFST.

Maximum length: MQ_CHANNEL_NAME_LENGTH.

Returned: Always.

ChannelType

Description: Channel Type. This specifies the type of channel for which the auto-definition has failed.

Identifier: MQIACH_CHANNEL_TYPE.

Data type: MQCFIN.

Values:

MQCHT_RECEIVER
Receiver.

MQCHT_SVRCONN
Server-connection (for use by clients).

MQCHT_CLUSSDR
Cluster-sender.

Returned: Always.

ErrorIdentifier

Description: Identifier of the cause of the error. This contains either the reason code (MQRC_* or MQRCCF_*) resulting from the channel definition attempt or the value MQRCCF_SUPPRESSED_BY_EXIT if the attempt to create the definition was disallowed by the exit.

Identifier: MQIACF_ERROR_IDENTIFIER.

Data type: MQCFIN.

Returned: Always.

ConnectionName

Description: Name of the partner attempting to establish connection.

Identifier: MQCACH_CONNECTION_NAME.

Data type: MQCFST.

Maximum length: MQ_CONN_NAME_LENGTH.

Returned: Always.

AuxErrorDataInt1

Description: Auxiliary error data. This contains the value returned by the exit in the *Feedback* field of the MQCXP to indicate why the auto definition has been disallowed.

Identifier: MQIACF_AUX_ERROR_DATA_INT_1.

Data type: MQCFIN.

Returned: Only if *ErrorIdentifier* contains MQRCCF_SUPPRESSED_BY_EXIT.

Channel Auto-definition OK

Event name:	Channel Auto-definition OK.
Reason code in MQCFH:	"2233 (08B9) (RC2233): MQRC_CHANNEL_AUTO_DEF_OK" on page 1238. Automatic channel definition succeeded.
Event description:	This condition is detected when the automatic definition of a channel is successful. The channel is defined by the MCA.
Event type:	Channel.
Platforms:	All, except IBM MQ for z/OS.
Event queue:	SYSTEM.ADMIN.CHANNEL.EVENT.

Event data

QMgrName

Description: Name of the queue manager generating the event.

Identifier: MQCA_Q_MGR_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: Always.

ChannelName

Description: Name of the channel being defined.

Identifier: MQCACH_CHANNEL_NAME.

Data type: MQCFST.

Maximum length: MQ_CHANNEL_NAME_LENGTH.

Returned: Always.

ChannelType

Description: Type of channel being defined.

Identifier: MQIACH_CHANNEL_TYPE.

Data type: MQCFIN.

Values:

MQCHT_RECEIVER
Receiver.

MQCHT_SVRCONN
Server-connection (for use by clients).

MQCHT_CLUSSDR
Cluster-sender.

Returned: Always.

ConnectionName

Description: Name of the partner attempting to establish connection.

Identifier: MQCACH_CONNECTION_NAME.

Data type: MQCFST.

Maximum length: MQ_CONN_NAME_LENGTH.

Returned: Always.

Channel Blocked

Event name:	Channel Blocked.
Reason code in MQCFH:	"2577 (0A11) (RC2577): MQRC_CHANNEL_BLOCKED" on page 1347 Channel blocked. "2578 (0A12) (RC2578): MQRC_CHANNEL_BLOCKED_WARNING" on page 1347 Channel blocked - warning mode.
Event description:	This event is issued when an attempt to start an inbound channel is blocked. For MQRC_CHANNEL_BLOCKED_WARNING, temporary access has been granted to the channel because the channel authentication record is defined with WARN set to YES.
Event type:	Channel.
Platforms:	All.
Event queue:	SYSTEM.ADMIN.CHANNEL.EVENT

Event data

QMgrName

Description: Name of the queue manager generating the event.

Identifier: MQCA_Q_MGR_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: Always.

Reason qualifier

Description: Identifier that qualifies the reason code

Identifier: MQIACF_REASON_QUALIFIER

Data type: MQCFIN.

Values:

MQRQ_CHANNEL_BLOCKED_ADDRESS
Channel was blocked due to its IP address being in the list to be refused

MQRQ_CHANNEL_BLOCKED_USERID
Channel was blocked due to its asserted or mapped user ID being in the list to be refused.

MQRQ_CHANNEL_BLOCKED_NOACCESS
Channel was blocked due to its IP address; TLS Peer name; remote queue manager name or client user ID being mapped to have no access.

Returned: Always.

ChannelName

Description: Channel Name.

Identifier: MQCACH_CHANNEL_NAME.

Data type: MQCFST.

Maximum length: MQ_CHANNEL_NAME_LENGTH.

Returned: If the Reason Qualifier is not MQRQ_CHANNEL_BLOCKED_ADDRESS. In that case the inbound connection is blocked before the channel name is known.

UserIdentifier

Description: User identifier that was blocked.

Identifier: MQCACF_USER_IDENTIFIER

Data type: MQCFST.

Maximum length: MQ_USER_ID_LENGTH

Returned: Only if the Reason Qualifier is MQRQ_CHANNEL_BLOCKED_USERID

ConnectionName

Description: Address of the partner attempting to establish connection

Identifier: MQCACH_CONNECTION_NAME.

Data type: MQCFST.

Maximum length: MQ_CONN_NAME_LENGTH.

Returned: Always

RemoteQMgrName

Description: Name of the partner queue manager attempting to establish connection.

Identifier: MQCA_REMOTE_Q_MGR_NAME

Data type: MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH

Returned: Only for inbound queue manager connections.

SSLPeerName

Description: The Distinguished Name in the certificate sent from the remote system.

Identifier: MQCACH_SSL_PEER_NAME

Data type: MQCFST.

Maximum length: MQ_DISTINGUISHED_NAME_LENGTH

Returned: Whenever the channel is using TLS and the client has not connected anonymously.

SSLIssuerName

Description: The Name of the Issuer in the certificate sent from the remote system.

Identifier: MQCA_SSL_CERT_ISSUER_NAME

Data type: MQCFST

Maximum length: MQ_DISTINGUISHED_NAME_LENGTH

Returned: Whenever the channel is using TLS and the client has not connected anonymously.

ClientUserIdentifier

Description: Client side user identifier of the partner attempting to establish connection.

Identifier: MQCACH_CLIENT_USER_ID

Data type: MQCFST.

Maximum length: MQ_USER_ID_LENGTH

Returned: Only for inbound client connections, if the Reason Qualifier is not MQRQ_CHANNEL_BLOCKED_ADDRESS. In that case the inbound connection is blocked before the client user Id name is known.

ApplType

Description: Type of application that made the API call.

Identifier: MQIA_APPL_TYPE

Data type: MQCFIN.

Returned: Only for inbound client connections. If the Reason Qualifier is not MQRQ_CHANNEL_BLOCKED_ADDRESS. In that case the inbound connection is blocked before the application name is known.

ApplName

Description: Name of the application that made the API call.

Identifier: MQCACF_APPL_NAME

Data type: MQCFST.

Maximum length: MQ_APPL_NAME_LENGTH

Returned: Only for inbound client connections. If the Reason Qualifier is not MQRQ_CHANNEL_BLOCKED_ADDRESS. In that case the inbound connection is blocked before the application name is known.

Channel Conversion Error

Event name:	Channel Conversion Error.
Reason code in MQCFH:	"2284 (08EC) (RC2284): MQRC_CHANNEL_CONV_ERROR" on page 1256. Channel conversion error.

Event description:	This condition is detected when a channel is unable to carry out data conversion and the MQGET call to get a message from the transmission queue resulted in a data conversion error. The reason for the failure is identified by *ConversionReasonCode*.
Event type:	Channel.
Platforms:	All.
Event queue:	SYSTEM.ADMIN.CHANNEL.EVENT.

Event data

QMgrName

Description: Name of the queue manager generating the event.

Identifier: MQCA_Q_MGR_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: Always.

ConversionReasonCode

Description: Identifier of the cause of the conversion error.

Identifier: MQIACF_CONV_REASON_CODE.

Data type: MQCFIN.

Values:

MQRC_CONVERTED_MSG_TOO_BIG (2120, X'848')
Converted message too big for application buffer.

MQRC_FORMAT_ERROR (2110, X'83E')
Message format not valid.

MQRC_NOT_CONVERTED (2119, X'847')
Application message data not converted.

MQRC_SOURCE_CCSID_ERROR (2111, X'83F')
Source coded character set identifier not valid.

MQRC_SOURCE_DECIMAL_ENC_ERROR (2113, X'841')
Packed-decimal encoding in message not recognized.

MQRC_SOURCE_FLOAT_ENC_ERROR (2114, X'842')
Floating-point encoding in message not recognized.

MQRC_SOURCE_INTEGER_ENC_ERROR (2112, X'840')
Integer encoding in message not recognized.

MQRC_TARGET_CCSID_ERROR (2115, X'843')
Target coded character set identifier not valid.

MQRC_TARGET_DECIMAL_ENC_ERROR (2117, X'845')
Packed-decimal encoding specified by receiver not recognized.

MQRC_TARGET_FLOAT_ENC_ERROR (2118, X'846')
Floating-point encoding specified by receiver not recognized.

MQRC_TARGET_INTEGER_ENC_ERROR (2116, X'844')
Integer encoding specified by receiver not recognized.

MQRC_TRUNCATED_MSG_ACCEPTED (2079, X'81F')
Truncated message returned (processing completed).

MQRC_TRUNCATED_MSG_FAILED (2080, X'820')
Truncated message returned (processing not completed).

Returned: Always.

ChannelName

Description: Channel name.

Identifier: MQCACH_CHANNEL_NAME.

Data type: MQCFST.

Maximum length: MQ_CHANNEL_NAME_LENGTH.

Returned: Always.

Format

Description: Format name.

Identifier: MQCACH_FORMAT_NAME.

Data type: MQCFST.

Maximum length: MQ_FORMAT_LENGTH.

Returned: Always.

XmitQName

Description: Transmission queue name.

Identifier: MQCACH_XMIT_Q_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_NAME_LENGTH.

Returned: Always.

ConnectionName

Description: If the channel has successfully established a TCP connection, this is the Internet address. Otherwise it is the contents of the *ConnectionName* field in the channel definition.

Identifier: MQCACH_CONNECTION_NAME.

Data type: MQCFST.

Maximum length: MQ_CONN_NAME_LENGTH.

Returned: Always.

Channel Not Activated

Event name: Channel Not Activated.

Reason code in MQCFH:	“2296 (08F8) (RC2296): MQRC_CHANNEL_NOT_ACTIVATED” on page 1260. Channel cannot be activated.
Event description:	This condition is detected when a channel is required to become active, either because it is starting, or because it is about to make another attempt to establish connection with its partner. However, it is unable to do so because the limit on the number of active channels has been reached. See the following: • Linux AIX **MaxActiveChannels** parameter in the `qm.ini` file for AIX and Linux. • Windows **MaxActiveChannels** parameter in the Registry for Windows. • z/OS **ACTCHL** parameter on the **ALTER QMGR** command for z/OS. The channel waits until it is able to take over an active slot released when another channel ceases to be active. At that time a Channel Activated event is generated.
Event type:	Channel.
Platforms:	All.
Event queue:	SYSTEM.ADMIN.CHANNEL.EVENT.

Event data

QMgrName

Description:	Name of the queue manager generating the event.
Identifier:	MQCA_Q_MGR_NAME.
Data type:	MQCFST.
Maximum length:	MQ_Q_MGR_NAME_LENGTH.
Returned:	Always.

ChannelName

Description:	Channel name.
Identifier:	MQCACH_CHANNEL_NAME.
Data type:	MQCFST.
Maximum length:	MQ_CHANNEL_NAME_LENGTH.
Returned:	Always.

XmitQName

Description:	Transmission queue name.
Identifier:	MQCACH_XMIT_Q_NAME.
Data type:	MQCFST.
Maximum length:	MQ_Q_NAME_LENGTH.
Returned:	For sender, server, cluster-sender, and cluster-receiver channel types only.

ConnectionName

Description:	If the channel has successfully established a TCP connection, this is the Internet address. Otherwise it is the contents of the `ConnectionName` field in the channel definition.
Identifier:	MQCACH_CONNECTION_NAME.
Data type:	MQCFST.
Maximum length:	MQ_CONN_NAME_LENGTH.
Returned:	Only for commands that do not contain a generic name.

Channel Not Available

Event name:	Channel Not Available.
Reason code in MQCFH:	"2537 (09E9) (RC2537): MQRC_CHANNEL_NOT_AVAILABLE" on page 1335. Channel not available.
Event description:	This is issued when an attempt to start an inbound channel is rejected.
Event type:	Channel.
Platforms:	All.
Event queue:	SYSTEM.ADMIN.CHANNEL.EVENT.

Event data

QMgrName

Description:	Name of the queue manager generating the event.
Identifier:	MQCA_Q_MGR_NAME.
Data type:	MQCFST.
Maximum length:	MQ_Q_MGR_NAME_LENGTH.
Returned:	Always.

ReasonQualifier

Description:	Identifier that qualifies the reason code.
Identifier:	MQIACF_REASON_QUALIFIER.
Data type:	MQCFIN.

Values:

MQRQ_MAX_ACTIVE_CHANNELS
Channel was unavailable due to maximum active channel instances (MaxActiveChannels qm.ini stanza on Multiplatforms or ACTCHL MQSC keyword on z/OS) limit being reached for the queue manager.

MQRQ_MAX_CHANNELS
Channel was unavailable due to maximum channel instances (MaxChannels qm.ini stanza on Multiplatforms or MAXCHL MQSC keyword on z/OS) limit being reached for the queue manager.

MQRQ_SVRCONN_INST_LIMIT
Channel was unavailable due to maximum active channel instances (MAXINST) limit being reached for the channel.

MQRQ_CLIENT_INST_LIMIT
Channel was unavailable due to maximum active channel instances (MAXINSTC) limit being reached for the client for the channel.

Returned: Always.

ChannelName

Description: Channel name.

Identifier: MQCACH_CHANNEL_NAME.

Data type: MQCFST.

Maximum length: MQ_CHANNEL_NAME_LENGTH.

Returned: Always.

ConnectionName

Description: Address of the partner attempting to establish connection.

Identifier: MQCACH_CONNECTION_NAME.

Data type: MQCFST.

Maximum length: MQ_CONN_NAME_LENGTH.

Returned: Always.

MaximumActiveChannels

Description: Maximum active channels.

Identifier: MQIA_ACTIVE_CHANNELS

Data type: MQCFIN.

Returned: Only where reason qualifier MQRQ_MAX_ACTIVE_CHANNELS.

MaximumChannels

Description: Maximum channels.

Identifier: MQIA_MAX_CHANNELS

Data type: MQCFIN

Returned: Only where reason qualifier MQRQ_MAX_CHANNELS.

MaximumInstances

Description: Maximum channel instances.

Identifier: MQIACH_MAX_INSTANCES

Data type: MQCFIN

Returned: Only where reason qualifier MQRQ_SVRCONN_INST_LIMIT.

MaximumClientInstances

Description: Maximum channel instances per client.

Identifier: MQIACH_MAX_INSTS_PER_CLIENT

Data type: MQCFIN

Returned: Only where reason qualifier MQRQ_CLIENT_INST_LIMIT.

Channel SSL Error

Event name:	Channel SSL Error.
Reason code in MQCFH:	"2371 (0943) (RC2371): MQRC_CHANNEL_SSL_ERROR" on page 1287. Channel SSL Error.
Event description:	This condition is detected when a channel using Transport Layer Security (TLS) fails to establish a connection. *ReasonQualifier* identifies the nature of the error.
Event type:	SSL.
Platforms:	All.
Event queue:	SYSTEM.ADMIN.CHANNEL.EVENT.

Event data

QMgrName

Description: Name of the queue manager generating the event.

Identifier: MQCA_Q_MGR_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: Always.

ReasonQualifier

Description: Identifier that qualifies the reason code.

Identifier: MQIACF_REASON_QUALIFIER.

Data type: MQCFIN.

Values:

MQRQ_SSL_HANDSHAKE_ERROR
The key exchange / authentication failure arose during the TLS handshake.

MQRQ_SSL_CIPHER_SPEC_ERROR
This error can mean any one of the following:

- The TLS client CipherSpec does not match that on the TLS server channel definition.
- An invalid CipherSpec has been specified.
- A CipherSpec has only been specified on one end of the TLS channel.

MQRQ_SSL_PEER_NAME_ERROR
The Distinguished Name in the certificate sent by one end of the TLS channel does not match the peer name on the end of the channel definition at the other end of the TLS channel.

MQRQ_SSL_CLIENT_AUTH_ERROR
The TLS server channel definition specified either SSLCAUTH(REQUIRED) or a SSLPEER value that was not blank, but the TLS client did not provide a certificate.

Returned: Always.

ChannelName

Description: Channel Name.

Identifier: MQCACH_CHANNEL_NAME.

Data type: MQCFST.

Maximum length: MQ_CHANNEL_NAME_LENGTH.

Returned: The *ChannelName* might not be available if the channel has not yet got far enough through its start-up process, in this case the channel name will not be returned. Otherwise always.

XmitQName

Description: Transmission queue name.

Identifier: MQCACH_XMIT_Q_NAME.

Data type: MQCFST.

Returned: For sender, server, cluster-sender and cluster-receiver channels only.

ConnectionName

Description: If the channel has successfully established a TCP connection, this is the Internet address. Otherwise it is the contents of the ConnectionName field in the channel definition.

Identifier: MQCACH_CONNECTION_NAME.

Data type: MQCFST.

Maximum length: MQ_CONN_NAME_LENGTH.

Returned: The *ConnectionName* might not be available if the channel has not yet got far enough through its start-up process, in this case the connection name will not be returned. Otherwise always.

SSLHandshakeStage

Description: Information about the TLS function call giving the error. For z/OS, details of function names can be found in the *System Secure Sockets Layer Programming Guide and Reference* SC24-5877.

Identifier: MQCACH_SSL_HANDSHAKE_STAGE.

Data type: MQCFST.

Maximum length: MQ_SSL_HANDSHAKE_STAGE_LENGTH.

Returned: This field is only present if *ReasonQualifier* is set to MQRQ_SSL_HANDSHAKE_ERROR.

SSLReturnCode

Description: A numeric return code from a failing TLS call.

Details of TLS Return Codes for specific platforms can be found as follows:

- z/OS For z/OS, see "Transport Layer Security (TLS) return codes for z/OS" on page 1143.
- Multi For Multiplatforms, see Transport Layer Security (TLS) return codes.

Identifier: MQIACH_SSL_RETURN_CODE.

Data type: MQCFIN.

Returned: This field is only present if *ReasonQualifier* is set to MQRQ_SSL_HANDSHAKE_ERROR.

SSLPeerName

Description: The Distinguished Name in the certificate sent from the remote system.

Identifier: MQCACH_SSL_PEER_NAME.

Data type: MQCFST.

Maximum length: MQ_DISTINGUISHED_NAME_LENGTH.

Returned: This field is only present if *ReasonQualifier* is set to MQRQ_SSL_PEER_NAME_ERROR and is not always present for this reason qualifier.

Channel SSL Warning

Event name:	Channel SSL Warning.
Reason code in MQCFH:	"2552 (09F8) (RC2552): MQRC_CHANNEL_SSL_WARNING" on page 1341. Channel SSL Warning.
Event description:	This condition is detected when a channel using Transport Layer Security (TLS) experiences a problem that does not cause it to fail to establish a TLS connection. *ReasonQualifier* identifies the nature of the event.
Event type:	SSL.
Platforms:	All.
Event queue:	SYSTEM.ADMIN.CHANNEL.EVENT.

Event data

QMgrName

Description: Name of the queue manager generating the event.

Identifier: MQCA_Q_MGR_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: Always.

ReasonQualifier

Description: Identifier that qualifies the reason code.

Identifier: MQIACF_REASON_QUALIFIER.

Data type: MQCFIN.

Values:

MQRQ_SSL_UNKNOWN_REVOCATION
An OCSP responder returned a response of Unknown. IBM MQ is configured to produce warnings but allow the connection to continue.

Returned: Always.

ChannelName

Description: Channel Name.

Identifier: MQCACH_CHANNEL_NAME.

Data type: MQCFST.

Maximum length: MQ_CHANNEL_NAME_LENGTH.

Returned: The *ChannelName* might not be available if the channel has not yet got far enough through its start-up process, in this case the channel name will not be returned. Otherwise always.

XmitQName

Description: Transmission queue name.

Identifier: MQCACH_XMIT_Q_NAME.

Data type: MQCFST.

Returned: For sender, server, cluster-sender and cluster-receiver channels only.

ConnectionName

Description: If the channel has successfully established a TCP connection, this is the Internet address. Otherwise it is the contents of the ConnectionName field in the channel definition.

Identifier: MQCACH_CONNECTION_NAME.

Data type: MQCFST.

Maximum length: MQ_CONN_NAME_LENGTH.

Returned: The *ConnectionName* may not be available if the channel has not yet got far enough through its start-up process, in this case the connection name will not be returned. Otherwise always.

Channel Started

Event name:	Channel Started.
Reason code in MQCFH:	"2282 (08EA) (RC2282): MQRC_CHANNEL_STARTED" on page 1256. Channel started.
Event description:	Either an operator has issued a Start Channel command, or an instance of a channel has been successfully established. This condition is detected when Initial Data negotiation is complete and resynchronization has been performed where necessary, such that message transfer can proceed.
Event type:	Channel.
Platforms:	All. Client connections do not produce this event.
Event queue:	SYSTEM.ADMIN.CHANNEL.EVENT.

Event data

QMgrName

Description:	Name of the queue manager generating the event.
Identifier:	MQCA_Q_MGR_NAME.
Data type:	MQCFST.
Maximum length:	MQ_Q_MGR_NAME_LENGTH.
Returned:	Always.

ChannelName

Description:	Channel name.
Identifier:	MQCACH_CHANNEL_NAME.
Data type:	MQCFST.
Maximum length:	MQ_CHANNEL_NAME_LENGTH.
Returned:	Always.

XmitQName

Description:	Transmission queue name.
Identifier:	MQCACH_XMIT_Q_NAME.
Data type:	MQCFST.
Maximum length:	MQ_Q_NAME_LENGTH.
Returned:	For sender, server, cluster-sender, and cluster-receiver channels only.

ConnectionName

Description:	If the channel has successfully established a TCP connection, this is the Internet address. Otherwise it is the contents of the *ConnectionName* field in the channel definition.
Identifier:	MQCACH_CONNECTION_NAME.
Data type:	MQCFST.
Maximum length:	MQ_CONN_NAME_LENGTH.

Returned: Only for commands that do not contain a generic name.

Channel Stopped

Event name:	Channel Stopped.
Reason code in MQCFH:	“2283 (08EB) (RC2283): MQRC_CHANNEL_STOPPED” on page 1256. Channel stopped.
Event description:	This is issued when a channel instance stops. It will only be issued if the channel instance previously issued a channel started event.
Event type:	Channel.
Platforms:	All. Client connections do not produce this event.
Event queue:	SYSTEM.ADMIN.CHANNEL.EVENT.

Event data

QMgrName

Description: Name of the queue manager generating the event.

Identifier: MQCA_Q_MGR_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: Always.

ReasonQualifier

Description: Identifier that qualifies the reason code.

Identifier: MQIACF_REASON_QUALIFIER.

Data type: MQCFIN.

Values:

MQRQ_CHANNEL_STOPPED_OK
Channel has been closed with either a zero return code or a warning return code.

MQRQ_CHANNEL_STOPPED_ERROR
Channel has been closed but there is an error reported and the channel is not in stopped or retry state.

MQRQ_CHANNEL_STOPPED_RETRY
Channel has been closed and it is in retry state.

MQRQ_CHANNEL_STOPPED_DISABLED
Channel has been closed and it is in a stopped state.

Returned: Always.

ChannelName

Description: Channel name.

Identifier: MQCACH_CHANNEL_NAME.

Data type: MQCFST.

Maximum length: MQ_CHANNEL_NAME_LENGTH.

Returned: Always.

ErrorIdentifier

Description: Identifier of the cause of the error. If a channel is stopped due to an error, this is the code that identifies the error. If the event message is because of a channel stop failure, the following fields are set:

1. *ReasonQualifier*, containing the value MQRQ_CHANNEL_STOPPED_ERROR
2. *ErrorIdentifier*, containing the code number of an error message that describes the error
3. *AuxErrorDataInt1*, containing error message integer insert 1
4. *AuxErrorDataInt2*, containing error message integer insert 2
5. *AuxErrorDataStr1*, containing error message string insert 1
6. *AuxErrorDataStr2*, containing error message string insert 2
7. *AuxErrorDataStr3*, containing error message string insert 3

The meanings of the error message inserts depend on the code number of the error message. Details of error-message code numbers and the inserts for specific platforms can be found as follows:

- z/OS For z/OS, see Distributed queuing message codes.
- Multi For Multiplatforms, the last four digits of *ErrorIdentifier* when displayed in hexadecimal notation indicate the decimal code number of the error message.

  For example, if *ErrorIdentifier* has the value X'xxxxyyyy', the message code of the error message explaining the error is AMQyyyy. See "AMQ messages on Multiplatforms" on page 244 for a description of these error messages.

Identifier: MQIACF_ERROR_IDENTIFIER.

Data type: MQCFIN.

Returned: Always.

AuxErrorDataInt1

Description: First integer of auxiliary error data for channel errors. If a channel is stopped due to an error, this is the first integer parameter that qualifies the error. This information is for use by IBM service personnel; include it in any problem report that you submit to IBM regarding this event message.

Identifier: MQIACF_AUX_ERROR_DATA_INT_1.

Data type: MQCFIN.

Returned: Always.

AuxErrorDataInt2

Description: Second integer of auxiliary error data for channel errors. If a channel is stopped due to an error, this is the second integer parameter that qualifies the error. This information is for use by IBM service personnel; include it in any problem report that you submit to IBM regarding this event message.

Identifier: MQIACF_AUX_ERROR_DATA_INT_2.

Data type: MQCFIN.

Returned: Always.

AuxErrorDataStr1

Description: First string of auxiliary error data for channel errors. If a channel is stopped due to an error, this is the first string parameter that qualifies the error. This information is for use by IBM service personnel; include it in any problem report that you submit to IBM regarding this event message.

Identifier: MQCACF_AUX_ERROR_DATA_STR_1.

Data type: MQCFST.

Returned: Always.

AuxErrorDataStr2

Description: Second string of auxiliary error data for channel errors. If a channel is stopped due to an error, this is the second string parameter that qualifies the error. This information is for use by IBM service personnel; include it in any problem report that you submit to IBM regarding this event message.

Identifier: MQCACF_AUX_ERROR_DATA_STR_2.

Data type: MQCFST.

Returned: Always.

AuxErrorDataStr3

Description: Third string of auxiliary error data for channel errors. If a channel is stopped due to an error, this is the third string parameter that qualifies the error. This information is for use by IBM service personnel; include it in any problem report that you submit to IBM regarding this event message.

Identifier: MQCACF_AUX_ERROR_DATA_STR_3.

Data type: MQCFST.

Returned: Always.

XmitQName

Description: Transmission queue name.

Identifier: MQCACH_XMIT_Q_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_NAME_LENGTH.

Returned: For sender, server, cluster-sender, and cluster-receiver channels only.

ConnectionName

Description: If the channel has successfully established a TCP connection, this is the Internet address. Otherwise it is the contents of the *ConnectionName* field in the channel definition.

Identifier: MQCACH_CONNECTION_NAME.

Data type: MQCFST or MQCFSL.

Maximum length: MQ_CONN_NAME_LENGTH.

Returned: Only for commands that do not contain a generic name.

Channel Stopped By User

Event name:	Channel Stopped By User.
Reason code in MQCFH:	"2279 (08E7) (RC2279): MQRC_CHANNEL_STOPPED_BY_USER" on page 1255. Channel stopped by user.
Event description:	This is issued when a user issues a STOP CHL command. *ReasonQualifier* identifies the reasons for stopping.
Event type:	Channel.
Platforms:	All.
Event queue:	SYSTEM.ADMIN.CHANNEL.EVENT.

Event data

QMgrName

Description: Name of the queue manager generating the event.

Identifier: MQCA_Q_MGR_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: Always.

ReasonQualifier

Description: Identifier that qualifies the reason code.

Identifier: MQIACF_REASON_QUALIFIER.

Data type: MQCFIN.

Values:

MQRQ_CHANNEL_STOPPED_DISABLED
Channel has been closed and it is in a stopped state.

Returned: Always.

ChannelName

Description: Channel name.

Identifier: MQCACH_CHANNEL_NAME.

Data type: MQCFST.

Maximum length: MQ_CHANNEL_NAME_LENGTH.

Returned: Always.

Command

Event name:	Command.

Reason code in MQCFH:	"2412 (096C) (RC2412): MQRC_COMMAND_MQSC" on page 1298. MQSC command successfully issued, or, "2413 (096D) (RC2413): MQRC_COMMAND_PCF" on page 1299. PCF command successfully issued.
Event description:	Command successfully issued.
Event type:	Command.
Platforms:	All.
Event queue:	SYSTEM.ADMIN.COMMAND.EVENT.

Event data

The event data consists of two groups, *CommandContext* and *CommandData*.

CommandContext

Description:	PCF group containing the elements related to the context of the issued command.
Identifier:	MQGACF_COMMAND_CONTEXT.
Data type:	MQCFGR.
PCF elements in group:	• *EventUserId* • *EventSecurityId* • *EventOrigin* • *EventQMgr* • *EventAccountingToken* • *EventIdentityData* • *EventApplType* • *EventApplName* • *EventApplOrigin* • *Command*
Returned:	Always.

EventUserId

Description:	The user ID that issued the command or call that generated the event. (This is the same user ID that is used to check the authority to issue the command; for commands received from a queue, this is also the user identifier (UserIdentifier) from the MQMD of the command message).
Identifier:	MQCACF_EVENT_USER_ID.
Data type:	MQCFST.
Maximum length:	MQ_USER_ID_LENGTH.
Returned:	Always.

Windows ***EventSecurityId***

Description:	The security ID (an extension to the user ID) that issued the command or call that generated the event.
Identifier:	MQBACF_EVENT_SECURITY_ID.

Data type: MQCFBS.

Maximum length: MQ_SECURITY_ID_LENGTH.

Returned: Only on Windows.

EventOrigin

Description: The origin of the action causing the event.

Identifier: MQIACF_EVENT_ORIGIN.

Data type: MQCFIN.

Values:

MQEVO_CONSOLE
Console command.

MQEVO_INIT
Initialization input data set command.

MQEVO_MSG
Command message on SYSTEM.COMMAND.INPUT.

MQEVO_INTERNAL
Directly by queue manager.

MQEVO_OTHER
None of the above.

Returned: Always.

EventQMgr

Description: The queue manager where the command was entered. (The queue manager where the command is executed and that generates the event is in the MQMD of the event message).

Identifier: MQCACF_EVENT_Q_MGR.

Data type: MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: Always.

EventAccountingToken

Description: For commands received as a message (MQEVO_MSG), the accounting token (AccountingToken) from the MQMD of the command message.

Identifier: MQBACF_EVENT_ACCOUNTING_TOKEN.

Data type: MQCFBS.

Maximum length: MQ_ACCOUNTING_TOKEN_LENGTH.

Returned: Only if EventOrigin is MQEVO_MSG.

EventIdentityData

Description: For commands received as a message (MQEVO_MSG), application identity data (ApplIdentityData) from the MQMD of the command message.

Identifier: MQCACF_EVENT_APPL_IDENTITY.

Data type: MQCFST.

Maximum length: MQ_APPL_IDENTITY_DATA_LENGTH.

Returned: Only if EventOrigin is MQEVO_MSG.

EventApplType

Description: For commands received as a message (MQEVO_MSG), the type of application (PutApplType) from the MQMD of the command message.

Identifier: MQIACF_EVENT_APPL_TYPE.

Data type: MQCFIN.

Returned: Only if EventOrigin is MQEVO_MSG.

EventApplName

Description: For commands received as a message (MQEVO_MSG), the name of the application (PutApplName) from the MQMD of the command message.

Identifier: MQCACF_EVENT_APPL_NAME.

Data type: MQCFST.

Maximum length: MQ_APPL_NAME_LENGTH.

Returned: Only if EventOrigin is MQEVO_MSG.

EventApplOrigin

Description: For commands received as a message (MQEVO_MSG), the application origin data (ApplOriginData) from the MQMD of the command message.

Identifier: MQCACF_EVENT_APPL_ORIGIN.

Data type: MQCFST.

Maximum length: MQ_APPL_ORIGIN_DATA_LENGTH.

Returned: Only if EventOrigin is MQEVO_MSG.

Command

Description: The command code.

Identifier: MQIACF_COMMAND.

Data type: MQCFIN.

Values:

- If the event relates to a PCF command, then the value is that of the Command parameter in the MQCFH structure in the command message.
- If the event relates to an MQSC command, then the value is as follows:

MQCMD_ARCHIVE_LOG
ARCHIVE LOG

MQCMD_BACKUP_CF_STRUC
BACKUP CFSTRUCT

MQCMD_CHANGE_AUTH_INFO
ALTER AUTHINFO

MQCMD_CHANGE_BUFFER_POOL
ALTER BUFFPOOL

MQCMD_CHANGE_CF_STRUC
ALTER CFSTRUCT

MQCMD_CHANGE_CHANNEL
ALTER CHANNEL

MQCMD_CHANGE_COMM_INFO
ALTER COMMINFO

MQCMD_CHANGE_LISTENER
ALTER LISTENER

MQCMD_CHANGE_NAMELIST
ALTER NAMELIST

MQCMD_CHANGE_PAGE_SET
ALTER PSID

MQCMD_CHANGE_PROCESS
ALTER PROCESS

MQCMD_CHANGE_Q
ALTER QLOCAL/QREMOTE/QALIAS/QMODEL

MQCMD_CHANGE_Q_MGR
ALTER QMGR, DEFINE MAXSMSGS

MQCMD_CHANGE_SECURITY
ALTER SECURITY

MQCMD_CHANGE_SERVICE
ALTER SERVICE

MQCMD_CHANGE_STG_CLASS
ALTER STGCLASS

MQCMD_CHANGE_SUBSCRIPTION
ALTER SUBSCRIPTION

MQCMD_CHANGE_TOPIC
ALTER TOPIC

MQCMD_CHANGE_TRACE
ALTER TRACE

MQCMD_CLEAR_Q
CLEAR QLOCAL

MQCMD_CLEAR_TOPIC_STRING
CLEAR TOPICSTR

MQCMD_CREATE_AUTH_INFO
DEFINE AUTHINFO

MQCMD_CREATE_BUFFER_POOL
DEFINE BUFFPOOL

MQCMD_CREATE_CF_STRUC
DEFINE CFSTRUCT

MQCMD_CREATE_CHANNEL
DEFINE CHANNEL

MQCMD_CREATE_COMM_INFO
DEFINE COMMINFO

MQCMD_CREATE_LISTENER
DEFINE LISTENER

MQCMD_CREATE_NAMELIST
DEFINE NAMELIST

MQCMD_CREATE_PAGE_SET
DEFINE PSID

MQCMD_CREATE_PROCESS
DEFINE PROCESS

MQCMD_CREATE_Q
DEFINE QLOCAL/QREMOTE/QALIAS/QMODEL

MQCMD_CREATE_SERVICE
DEFINE SERVICE

MQCMD_CREATE_STG_CLASS
DEFINE STGCLASS

MQCMD_CREATE_SUBSCRIPTION
DEFINE SUB

MQCMD_CREATE_TOPIC
DEFINE TOPIC

MQCMD_DELETE_AUTH_INFO
DELETE AUTHINFO

MQCMD_DELETE_CF_STRUC
DELETE CFSTRUCT

MQCMD_DELETE_CHANNEL
DELETE CHANNEL

MQCMD_DELETE_COMM_INFO
DELETE COMMINFO

MQCMD_DELETE_LISTENER
DELETE LISTENER

MQCMD_DELETE_NAMELIST
DELETE NAMELIST

MQCMD_DELETE_PAGE_SET
DELETE PSID

MQCMD_DELETE_PROCESS
DELETE PROCESS

MQCMD_DELETE_Q
DELETE QLOCAL/QREMOTE/QALIAS/QMODEL

MQCMD_DELETE_SERVICE
DELETE SERVICE

MQCMD_DELETE_STG_CLASS
DELETE STGCLASS

MQCMD_DELETE_SUBSCRIPTION
DELETE SUBSCRIPTION

MQCMD_DELETE_TOPIC
DELETE TOPIC

MQCMD_INQUIRE_ARCHIVE
DISPLAY ARCHIVE

MQCMD_INQUIRE_AUTH_INFO
DISPLAY AUTHINFO

MQCMD_INQUIRE_CF_STRUC
DISPLAY CFSTRUCT

MQCMD_INQUIRE_CF_STRUC_STATUS
DISPLAY CFSTATUS

MQCMD_INQUIRE_CHANNEL
DISPLAY CHANNEL

MQCMD_INQUIRE_CHANNEL_INIT
DISPLAY CHINIT

MQCMD_INQUIRE_CHANNEL_STATUS
DISPLAY CHSTATUS

MQCMD_INQUIRE_CHLAUTH_RECS
DISPLAY CHLAUTH

MQCMD_INQUIRE_CLUSTER_Q_MGR
DISPLAY CLUSQMGR

MQCMD_INQUIRE_CMD_SERVER
DISPLAY CMDSERV

MQCMD_INQUIRE_COMM_INFO
DISPLAY COMMINFO

MQCMD_INQUIRE_CONNECTION
DISPLAY CONN

MQCMD_INQUIRE_LISTENER
DISPLAY LISTENER

MQCMD_INQUIRE_LOG
DISPLAY LOG

MQCMD_INQUIRE_NAMELIST
DISPLAY NAMELIST

MQCMD_INQUIRE_PROCESS
DISPLAY PROCESS

MQCMD_INQUIRE_PUBSUB_STATUS
DISPLAY PUBSUB

MQCMD_INQUIRE_Q
DISPLAY QUEUE

MQCMD_INQUIRE_Q_MGR
DISPLAY QMGR, DISPLAY MAXSMSGS

MQCMD_INQUIRE_QSG
DISPLAY GROUP

MQCMD_INQUIRE_Q_STATUS
DISPLAY QSTATUS

MQCMD_INQUIRE_SECURITY
DISPLAY SECURITY

MQCMD_INQUIRE_SERVICE
DISPLAY SERVICE

MQCMD_INQUIRE_STG_CLASS
DISPLAY STGCLASS

MQCMD_INQUIRE_SUBSCRIPTION
DISPLAY SUB

MQCMD_INQUIRE_SUB_STATUS
DISPLAY SBSTATUS

MQCMD_INQUIRE_SYSTEM
DISPLAY SYSTEM

MQCMD_INQUIRE_THREAD
DISPLAY THREAD

MQCMD_INQUIRE_TOPIC
DISPLAY TOPIC

MQCMD_INQUIRE_TOPIC_STATUS
DISPLAY TPSTATUS

MQCMD_INQUIRE_TRACE
DISPLAY TRACE

MQCMD_INQUIRE_USAGE
DISPLAY USAGE

MQCMD_MOVE_Q
MOVE QLOCAL

MQCMD_PING_CHANNEL
PING CHANNEL

MQCMD_RECOVER_BSDS
RECOVER BSDS

MQCMD_RECOVER_CF_STRUC
RECOVER CFSTRUCT

MQCMD_REFRESH_CLUSTER
REFRESH CLUSTER

MQCMD_REFRESH_Q_MGR
REFRESH QMGR

MQCMD_REFRESH_SECURITY
REFRESH SECURITY

MQCMD_RESET_CHANNEL
RESET CHANNEL

MQCMD_RESET_CLUSTER
RESET CLUSTER

MQCMD_RESET_Q_MGR
RESET QMGR

MQCMD_RESET_Q_STATS
RESET QSTATS

MQCMD_RESET_TPIPE
RESET TPIPE

MQCMD_RESOLVE_CHANNEL
RESOLVE CHANNEL

MQCMD_RESOLVE_INDOUBT
RESOLVE INDOUBT

MQCMD_RESUME_Q_MGR
RESUME QMGR other than CLUSTER/CLUSNL

MQCMD_RESUME_Q_MGR_CLUSTER
RESUME QMGR CLUSTER/CLUSNL

MQCMD_REVERIFY_SECURITY
REVERIFY SECURITY

MQCMD_SET_ARCHIVE
SET ARCHIVE

MQCMD_SET_CHLAUTH_REC
SET CHLAUTH

MQCMD_SET_LOG
SET LOG

MQCMD_SET_SYSTEM
SET SYSTEM

MQCMD_START_CHANNEL
START CHANNEL

MQCMD_START_CHANNEL_INIT
START CHINIT

MQCMD_START_CHANNEL_LISTENER
START LISTENER

MQCMD_START_CMD_SERVER
START CMDSERV

MQCMD_START_SERVICE
START SERVICE

MQCMD_START_TRACE
START TRACE

MQCMD_STOP_CHANNEL
STOP CHANNEL

MQCMD_STOP_CHANNEL_INIT
STOP CHINIT

MQCMD_STOP_CHANNEL_LISTENER
STOP LISTENER

MQCMD_STOP_CMD_SERVER
STOP CMDSERV

MQCMD_STOP_CONNECTION
STOP CONN

MQCMD_STOP_SERVICE
STOP SERVICE

MQCMD_STOP_TRACE
STOP TRACE

MQCMD_SUSPEND_Q_MGR
SUSPEND QMGR other than CLUSTER/CLUSNL

MQCMD_SUSPEND_Q_MGR_CLUSTER
SUSPEND QMGR CLUSTER/CLUSNL

Returned: Always.

CommandData

Description: PCF group containing the elements related to the command data.

Identifier: MQGACF_COMMAND_DATA.

Data type: MQCFGR.

PCF elements in group:
- If generated for an MQSC command, this group only contains the PCF element *CommandMQSC*.
- If generated for a PCF command, this group contains the PCF elements that make up the PCF command, exactly as in the command message.

Returned: Always.

CommandMQSC

Description: The text of the MQSC command.

Identifier: MQCACF_COMMAND_MQSC.

Data type: MQCFST.

Maximum length: MQ_COMMAND_MQSC_LENGTH.

Returned: Only if Reason in the message descriptor is MQRC_COMMAND_MQSC.

Create object

Event name:	Create object.
Reason code in MQCFH:	"2367 (093F) (RC2367): MQRC_CONFIG_CREATE_OBJECT" on page 1285. New object created.
Event description:	A DEFINE or DEFINE REPLACE command was issued which successfully created a new object.
Event type:	Configuration.
Platforms:	All.
Event queue:	SYSTEM.ADMIN.CONFIG.EVENT.

Event data

EventUserId

Description: The user ID that issued the command or call that generated the event. (This is the same user ID that is used to check the authority to issue the command or call; for commands received from a queue, this is also the user identifier (UserIdentifier) from the MQMD of the command message).

Identifier: MQCACF_EVENT_USER_ID.

Data type: MQCFST.

Maximum length: MQ_USER_ID_LENGTH.

Returned: Always.

EventOrigin

Description: The origin of the action causing the event.

Identifier: MQIACF_EVENT_ORIGIN.

Data type: MQCFIN.

Values:

MQEVO_CONSOLE
Console command.

MQEVO_INIT
Initialization input data set command.

MQEVO_INTERNAL
Directly by queue manager.

MQEVO_MQSET
MQSET call.

MQEVO_MSG
Command message on SYSTEM.COMMAND.INPUT.

MQEVO_OTHER
None of the above.

Returned: Always.

EventQMgr

Description: The queue manager where the command or call was entered. (The queue manager where the command is executed and that generates the event is in the MQMD of the event message).

Identifier: MQCACF_EVENT_Q_MGR.

Data type: MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: Always.

EventAccountingToken

Description: For commands received as a message (MQEVO_MSG), the accounting token (AccountingToken) from the MQMD of the command message.

Identifier: MQBACF_EVENT_ACCOUNTING_TOKEN.

Data type: MQCFBS.

Maximum length: MQ_ACCOUNTING_TOKEN_LENGTH.

Returned: Only if EventOrigin is MQEVO_MSG.

EventApplIdentity

Description: For commands received as a message (MQEVO_MSG), application identity data (ApplIdentityData) from the MQMD of the command message.

Identifier: MQCACF_EVENT_APPL_IDENTITY.

Data type: MQCFST.

Maximum length: MQ_APPL_IDENTITY_DATA_LENGTH.

Returned: Only if EventOrigin is MQEVO_MSG.

EventApplType

Description: For commands received as a message (MQEVO_MSG), the type of application (PutApplType) from the MQMD of the command message.

Identifier: MQIACF_EVENT_APPL_TYPE.

Data type: MQCFIN.

Returned: Only if EventOrigin is MQEVO_MSG.

EventApplName

Description: For commands received as a message (MQEVO_MSG), the name of the application (PutApplName) from the MQMD of the command message.

Identifier: MQCACF_EVENT_APPL_NAME.

Data type: MQCFST.

Maximum length: MQ_APPL_NAME_LENGTH.

Returned: Only if EventOrigin is MQEVO_MSG.

EventApplOrigin

Description: For commands received as a message (MQEVO_MSG), the application origin data (ApplOriginData) from the MQMD of the command message.

Identifier: MQCACF_EVENT_APPL_ORIGIN.

Data type: MQCFST.

Maximum length: MQ_APPL_ORIGIN_DATA_LENGTH.

Returned: Only if EventOrigin is MQEVO_MSG.

ObjectType

Description: Object type:

Identifier: MQIACF_OBJECT_TYPE.

Data type: MQCFIN.

Values:

MQOT_CHANNEL
Channel.

MQOT_CHLAUTH
Channel authentication record.

MQOT_NAMELIST
Namelist.

MQOT_NONE
No object.

MQOT_PROCESS
Process.

MQOT_Q
Queue.

MQOT_STORAGE_CLASS
Storage class.

MQOT_AUTH_INFO
Authentication information.

MQOT_CF_STRUC
CF structure.

MQOT_TOPIC
Topic.

MQOT_COMM_INFO
Communication information.

MQOT_LISTENER
Channel Listener.

Returned: Always.

ObjectName

Description: Object name:

Identifier : Identifier will be according to object type.

- MQCACH_CHANNEL_NAME
- MQCA_NAMELIST_NAME
- MQCA_PROCESS_NAME
- MQCA_Q_NAME
- MQCA_STORAGE_CLASS
- MQCA_AUTH_INFO_NAME
- MQCA_CF_STRUC_NAME
- MQCA_TOPIC_NAME
- MQCA_COMM_INFO_NAME
- MQCACH_LISTENER_NAME

Note: MQCACH_CHANNEL_NAME can also be used for channel authentication.

Data type: MQCFST.

Maximum length: MQ_OBJECT_NAME_LENGTH.

Returned: Always

Disposition

Description: Object disposition:

Identifier: MQIA_QSG_DISP.

Data type: MQCFIN.

Values:

MQQSGD_Q_MGR
Object resides on page set of queue manager.

MQQSGD_SHARED
Object resides in shared repository and messages are shared in coupling facility.

MQQSGD_GROUP
Object resides in shared repository.

MQQSGD_COPY
Object resides on page set of queue manager and is a local copy of a GROUP object.

Returned: Always, except for CF structure objects.

Object attributes

A parameter structure is returned for each attribute of the object. The attributes returned depend on the object type. For more information see "Object attributes for event data" on page 74

Default Transmission Queue Type Error

Event name:	Default Transmission Queue Type Error.
Reason code in MQCFH:	"2198 (0896) (RC2198): MQRC_DEF_XMIT_Q_TYPE_ERROR" on page 1230. Default transmission queue not local.
Event description:	An MQOPEN or MQPUT1 call was issued specifying a remote queue as the destination. Either a local definition of the remote queue was specified, or a queue manager alias was being resolved, but in either case the **XmitQName** attribute in the local definition is blank. No transmission queue is defined with the same name as the destination queue manager, so the local queue manager has attempted to use the default transmission queue. However, although there is a queue defined by the **DefXmitQName** queue manager attribute, it is not a local queue.
Event type:	Remote.
Platforms:	All.
Event queue:	SYSTEM.ADMIN.QMGR.EVENT.

Event data

QMgrName

Description: Name of the queue manager generating the event.

Identifier: MQCA_Q_MGR_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: Always.

QName

Description: Queue name from object descriptor (MQOD).

Identifier: MQCA_Q_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_NAME_LENGTH.

Returned: Always.

XmitQName

Description: Default transmission queue name.

Identifier: MQCA_XMIT_Q_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_NAME_LENGTH.

Returned: Always.

QType

Description: Type of default transmission queue.

Identifier: MQIA_Q_TYPE.

Data type: MQCFIN.

Values:

MQQT_ALIAS
Alias queue definition.

MQQT_REMOTE
Local definition of a remote queue.

Returned: Always.

ApplType

Description: Type of application making the MQI call that caused the event.

Identifier: MQIA_APPL_TYPE.

Data type: MQCFIN.

Returned: Always.

ApplName

Description: Name of the application making the MQI call that caused the event.

Identifier: MQCACF_APPL_NAME.

Data type: MQCFST.

Maximum length: MQ_APPL_NAME_LENGTH.

Returned: Always.

ObjectQMgrName

Description: Name of the object queue manager.

Identifier: MQCACF_OBJECT_Q_MGR_NAME.

Data type:	MQCFST.
Maximum length:	MQ_Q_MGR_NAME_LENGTH.
Returned:	If the *ObjectName* in the object descriptor (MQOD), when the object was opened, is not the queue manager currently connected.

ConnName

Description:	Connection name for client connection.
Identifier:	MQCACH_CONNECTION_NAME.
Data type:	MQCFST.
Maximum length:	MQ_CONN_NAME_LENGTH.
Returned:	If the application making the MQI call that caused the event is a client attached application.

ChannelName

Description:	Channel name for client connection.
Identifier:	MQCACH_CHANNEL_NAME.
Data type:	MQCFST.
Maximum length:	MQ_CHANNEL_NAME_LENGTH
Returned:	If the application making the MQI call that caused the event is a client attached application.

Related tasks

Defining system objects

Related reference

Sample definitions supplied with IBM MQ

DefXmitQName (MQCHAR48)

DefXmitQName (48-byte character string)

Default Transmission Queue Usage Error

Event name:	Default Transmission Queue Usage Error.
Reason code in MQCFH:	"2199 (0897) (RC2199): MQRC_DEF_XMIT_Q_USAGE_ERROR" on page 1230. Default transmission queue usage error.
Event description:	An MQOPEN or MQPUT1 call was issued specifying a remote queue as the destination. Either a local definition of the remote queue was specified, or a queue manager alias was being resolved, but in either case the **XmitQName** attribute in the local definition is blank. No transmission queue is defined with the same name as the destination queue manager, so the local queue manager has attempted to use the default transmission queue. However, the queue defined by the **DefXmitQName** queue manager attribute does not have a **Usage** attribute of MQUS_TRANSMISSION.
Event type:	Remote.
Platforms:	All.
Event queue:	SYSTEM.ADMIN.QMGR.EVENT.

Event data

QMgrName

Description: Name of the queue manager generating the event.
Identifier: MQCA_Q_MGR_NAME.
Data type: MQCFST.
Maximum length: MQ_Q_MGR_NAME_LENGTH.
Returned: Always.

QName

Description: Queue name from object descriptor (MQOD).
Identifier: MQCA_Q_NAME.
Data type: MQCFST.
Maximum length: MQ_Q_NAME_LENGTH.
Returned: Always.

XmitQName

Description: Default transmission queue name.
Identifier: MQCA_XMIT_Q_NAME.
Data type: MQCFST.
Maximum length: MQ_Q_NAME_LENGTH.
Returned: Always.

ApplType

Description: Type of application making the MQI call that caused the event.
Identifier: MQIA_APPL_TYPE.
Data type: MQCFIN.
Returned: Always.

ApplName

Description: Name of the application making the MQI call that caused the event.
Identifier: MQCACF_APPL_NAME.
Data type: MQCFST.
Maximum length: MQ_APPL_NAME_LENGTH.
Returned: Always.

ObjectQMgrName

Description: Name of the object queue manager.
Identifier: MQCACF_OBJECT_Q_MGR_NAME.
Data type: MQCFST.
Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned:	If the *ObjectName* in the object descriptor (MQOD), when the object was opened, is not the queue manager currently connected.

ConnName

Description:	Connection name for client connection.
Identifier:	MQCACH_CONNECTION_NAME.
Data type:	MQCFST.
Maximum length:	MQ_CONN_NAME_LENGTH.
Returned:	If the application making the MQI call that caused the event is a client attached application.

ChannelName

Description:	Channel name for client connection.
Identifier:	MQCACH_CHANNEL_NAME.
Data type:	MQCFST.
Maximum length:	MQ_CHANNEL_NAME_LENGTH
Returned:	If the application making the MQI call that caused the event is a client attached application.

Related tasks

Defining system objects

Related reference

Sample definitions supplied with IBM MQ

DefXmitQName (MQCHAR48)

DefXmitQName (48-byte character string)

Delete Authority Record

Event name:	Delete Authority Record
Reason code in MQCFH:	"2369 (0941) (RC2369): MQRC_CONFIG_DELETE_OBJECT" on page 1286. Object deleted.
Event description:	A Delete Authority Record command was issued, or an object was deleted, which successfully deleted an authority record.
Event type:	Configuration
Platforms:	All except z/OS.
Event queue:	SYSTEM.ADMIN.CONFIG.EVENT.

Event data

EventQMgr

Description:	The queue manager where the command or call was entered. That is, the queue manager where the command is processed and that generates the event is in the MQMD of the event message.
Identifier:	MQCACF_EVENT_Q_MGR
Data type:	MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: Always.

EventUserId

Description: The user ID that issued the command or call that generated the event.

This is the same user ID that is used to check the authority to issue the command or call; for commands received from a queue, this is also the user identifier (**UserIdentifier**) from the message descriptor of the command message.

.

Identifier: MQCACF_EVENT_USER_ID

Data type: MQCFST.

Maximum length: MQ_USER_ID_LENGTH.

Returned: Always.

EventOrigin

Description: The origin of the action causing the event.

Identifier: MQIACF_EVENT_ORIGIN

Data type: MQCFIN.

Values:

MQEVO_CONSOLE
Console command (runmqsc or setmqaut)

MQEVO_INTERNAL
Directly by queue manager

MQEVO_MSG
Command message on SYSTEM.ADMIN.COMMAND.QUEUE

Returned: Always

EventAccountingToken

Description: For commands received as a message (MQEVO_MSG), the accounting token (**AccountingToken**) from the message descriptor of the command message.

Identifier: MQBACF_EVENT_ACCOUNTING_TOKEN

Data type: MQCFBS

Maximum length: MQ_ACCOUNTING_TOKEN_LENGTH

Returned: Only if **EventOrigin** is MQEVO_MSG.

EventApplIdentity

Description: For commands received as a message (MQEVO_MSG), application identity data (**ApplIdentityData**) from the message descriptor of the command message.

Identifier: MQMQCACF_EVENT_APPL_IDENTITY

Data type: MQCFST

Maximum length: MQ_APPL_IDENTITY_DATA_LENGTH

Returned: Only if **EventOrigin** is MQEVO_MSG.

EventApplType

Description: For commands received as a message (MQEVO_MSG), the type of application (**PutApplType**) from the message descriptor of the command message.

Identifier: MQIACF_EVENT_APPL_TYPE

Data type: MQCFIN

Values:

Returned: Only if **EventOrigin** is MQEVO_MSG.

EventApplName

Description: For commands received as a message (MQEVO_MSG), the name of the application (**PutApplName**) from the message descriptor of the command message.

Identifier: MQCACF_EVENT_APPL_NAME

Data type: MQCFST

Maximum length: MQ_APPL_NAME_LENGTH

Returned: Only if **EventOrigin** is MQEVO_MSG.

EventApplOrigin

Description: For commands received as a message (MQEVO_MSG), the application origin data (**ApplOriginData**) from the message descriptor of the command message.

Identifier: MQCACF_EVENT_APPL_ORIGIN

Data type: MQCFST

Maximum length: MQ_APPL_ORIGIN_DATA_LENGTH

Returned: Only if **EventOrigin** is MQEVO_MSG.

ObjectType

Description: Object type

Identifier: MQIACF_OBJECT_TYPE

Data type: MQCFIN

Values: MQOT_AUTH_REC

Returned: Always

ProfileName

Description: Object or generic profile name

Identifier: MQCACF_AUTH_PROFILE_NAME

Data type: MQCFST

Maximum length: MQ_AUTH_PROFILE_NAME_LENGTH

Returned: Always

Object attributes

A parameter structure is returned for each attribute of the object. The attributes returned depend on the object type. For more information see “Authority Record attributes” on page 119.

Delete object

Event name:	Delete object.
Reason code in MQCFH:	“2369 (0941) (RC2369): MQRC_CONFIG_DELETE_OBJECT” on page 1286. Object deleted.
Event description:	A DELETE command or MQCLOSE call was issued that successfully deleted an object.
Event type:	Configuration.
Platforms:	All.
Event queue:	SYSTEM.ADMIN.CONFIG.EVENT.

Event data

EventUserId

Description: The user ID that issued the command or call that generated the event. (This is the same user ID that is used to check the authority to issue the command or call; for commands received from a queue, this is also the user identifier (UserIdentifier) from the MQMD of the command message).

Identifier: MQCACF_EVENT_USER_ID.

Data type: MQCFST.

Maximum length: MQ_USER_ID_LENGTH.

Returned: Always.

EventOrigin

Description: The origin of the action causing the event.

Identifier: MQIACF_EVENT_ORIGIN.

Data type: MQCFIN.

Values:

MQEVO_CONSOLE
Console command.

MQEVO_INIT
Initialization input data set command.

MQEVO_INTERNAL
Directly by queue manager.

MQEVO_MSG
Command message on SYSTEM.COMMAND.INPUT.

MQEVO_OTHER
None of the above.

Returned: Always.

EventQMgr

Description: The queue manager where the command or call was entered. (The queue manager where the command is executed and that generates the event is in the MQMD of the event message).

Identifier: MQCACF_EVENT_Q_MGR.

Data type: MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: Always.

EventAccountingToken

Description: For commands received as a message (MQEVO_MSG), the accounting token (AccountingToken) from the MQMD of the command message.

Identifier: MQBACF_EVENT_ACCOUNTING_TOKEN.

Data type: MQCFBS.

Maximum length: MQ_ACCOUNTING_TOKEN_LENGTH.

Returned: Only if EventOrigin is MQEVO_MSG.

EventApplIdentity

Description: For commands received as a message (MQEVO_MSG), application identity data (ApplIdentityData) from the MQMD of the command message.

Identifier: MQCACF_EVENT_APPL_IDENTITY.

Data type: MQCFST.

Maximum length: MQ_APPL_IDENTITY_DATA_LENGTH.

Returned: Only if EventOrigin is MQEVO_MSG.

EventApplType

Description: For commands received as a message (MQEVO_MSG), the type of application (PutApplType) from the MQMD of the command message.

Identifier: MQIACF_EVENT_APPL_TYPE.

Data type: MQCFIN.

Returned: Only if EventOrigin is MQEVO_MSG.

EventApplName

Description: For commands received as a message (MQEVO_MSG), the name of the application (PutApplName) from the MQMD of the command message.

Identifier: MQCACF_EVENT_APPL_NAME.

Data type: MQCFST.

Maximum length: MQ_APPL_NAME_LENGTH.

Returned: Only if EventOrigin is MQEVO_MSG.

EventApplOrigin

Description: For commands received as a message (MQEVO_MSG), the application origin data (ApplOriginData) from the MQMD of the command message.

Identifier: MQCACF_EVENT_APPL_ORIGIN.

Data type: MQCFST.

Maximum length: MQ_APPL_ORIGIN_DATA_LENGTH.

Returned: Only if EventOrigin is MQEVO_MSG.

ObjectType

Description: Object type:

Identifier: MQIACF_OBJECT_TYPE.

Data type: MQCFIN.

Values:

MQOT_CHANNEL
Channel.

MQOT_CHLAUTH
Channel authentication record.

MQOT_NAMELIST
Namelist.

MQOT_NONE
No object.

MQOT_PROCESS
Process.

MQOT_Q
Queue.

MQOT_STORAGE_CLASS
Storage class.

MQOT_AUTH_INFO
Authentication information.

MQOT_CF_STRUC
CF structure.

MQOT_TOPIC
Topic.

MQOT_COMM_INFO
Communication information.

MQOT_LISTENER
Channel Listener.

Returned: Always.

ObjectName

Description: Object name:

Identifier : Identifier will be according to object type.

- MQCACH_CHANNEL_NAME
- MQCA_NAMELIST_NAME
- MQCA_PROCESS_NAME
- MQCA_Q_NAME
- MQCA_STORAGE_CLASS
- MQCA_AUTH_INFO_NAME
- MQCA_CF_STRUC_NAME
- MQCA_TOPIC_NAME
- MQCA_COMM_INFO_NAME
- MQCACH_LISTENER_NAME

Note: MQCACH_CHANNEL_NAME can also be used for channel authentication.

Data type: MQCFST.

Maximum length: MQ_OBJECT_NAME_LENGTH.

Returned: Always

Disposition

Description: Object disposition:

Identifier: MQIA_QSG_DISP.

Data type: MQCFIN.

Values:

MQQSGD_Q_MGR
Object resides on page set of queue manager.

MQQSGD_SHARED
Object resides in shared repository and messages are shared in coupling facility.

MQQSGD_GROUP
Object resides in shared repository.

MQQSGD_COPY
Object resides on page set of queue manager and is a local copy of a GROUP object.

Returned: Always, except for CF structure objects.

Object attributes

A parameter structure is returned for each attribute of the object. The attributes returned depend on the object type. For more information see "Object attributes for event data" on page 74.

Get Inhibited

Event name:	Get Inhibited.
Reason code in MQCFH:	"2016 (07E0) (RC2016): MQRC_GET_INHIBITED" on page 1161. Gets inhibited for the queue.
Event description:	MQGET calls are currently inhibited for the queue (see InhibitGet (MQLONG) for the **InhibitGet** queue attribute) or for the queue to which this queue resolves.
Event type:	Inhibit.
Platforms:	All.
Event queue:	SYSTEM.ADMIN.QMGR.EVENT.

Event data

QMgrName

Description: Name of the queue manager generating the event.

Identifier: MQCA_Q_MGR_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: Always.

QName

Description: Queue name from object descriptor (MQOD).

Identifier: MQCA_Q_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_NAME_LENGTH.

Returned: Always.

ApplType

Description: Type of application that issued the get.

Identifier: MQIA_APPL_TYPE.

Data type: MQCFIN.

Returned: Always.

ApplName

Description: Name of the application that issued the get.

Identifier: MQCACF_APPL_NAME.

Data type: MQCFST.

Maximum length: MQ_APPL_NAME_LENGTH.

Returned: Always.

ConnName

Description: Connection name for client connection.

Identifier: MQCACH_CONNECTION_NAME.

Data type: MQCFST.

Maximum length: MQ_CONN_NAME_LENGTH.

Returned: If the application making the MQI call that caused the event is a client attached application.

ChannelName

Description: Channel name for client connection.

Identifier: MQCACH_CHANNEL_NAME.

Data type: MQCFST.

Maximum length: MQ_CHANNEL_NAME_LENGTH

Returned: If the application making the MQI call that caused the event is a client attached application.

Related tasks

Setting queue attributes

Related reference

IBM i InhibitGet (10-digit signed integer)

Logger

Event name:	Logger.
Reason code in MQCFH:	"2411 (096B) (RC2411): MQRC_LOGGER_STATUS" on page 1298. New log extent started.
Event description:	Issued when a queue manager starts writing to a new log extent IBM i or on IBM i a new journal receiver.
Event type:	Logger.
Platforms:	All, except IBM MQ for z/OS.
Event queue:	SYSTEM.ADMIN.LOGGER.EVENT.

Event data

QMgrName

Description:	Name of the queue manager generating the event.
Identifier:	MQCA_Q_MGR_NAME.
Data type:	MQCFST.
Maximum length:	MQ_Q_MGR_NAME_LENGTH.
Returned:	Always.

CurrentLogExtent

Description:	Name of the log extent IBM i, or on IBM i the journal receiver being written, when the event message was generated.
Identifier:	MQCACF_CURRENT_LOG_EXTENT_NAME.
Data type:	MQCFST.
Maximum length:	MQ_LOG_EXTENT_NAME_LENGTH.
Returned:	Always.

RestartRecoveryLogExtent

Description:	Name of the oldest log extent IBM i, or on IBM i the oldest journal receiver, required by the queue manager to perform restart recovery.
Identifier:	MQCACF_RESTART_LOG_EXTENT_NAME.
Data type:	MQCFST.
Maximum length:	MQ_LOG_EXTENT_NAME_LENGTH.
Returned:	Always.

MediaRecoveryLogExtent

Description:	Name of the oldest log extent IBM i, or on IBM i the oldest journal receiver, required by the queue manager to perform media recovery.
Identifier:	MQCACF_MEDIA_LOG_EXTENT_NAME.
Data type:	MQCFST.

Maximum length: MQ_LOG_EXTENT_NAME_LENGTH.

Returned: Always.

LogPath

Description: The directory where log files are created by the queue manager.

Identifier: MQCACF_LOG_PATH.

Data type: MQCFST.

Maximum length: MQ_LOG_PATH_LENGTH.

Returned: Always.

Multi *Not Authorized (type 1)*

Event name:	Not Authorized (type 1).
Reason code in MQCFH:	"2035 (07F3) (RC2035): MQRC_NOT_AUTHORIZED" on page 1169. Not authorized for access.
Event description:	On an MQCONN or system connection call, the user is not authorized to connect to the queue manager. *ReasonQualifier* identifies the nature of the error.
Event type:	Authority.
Platforms:	All, except IBM MQ for z/OS.
Event queue:	SYSTEM.ADMIN.QMGR.EVENT.

Event data

QMgrName

Description: Name of the queue manager generating the event.

Identifier: MQCA_Q_MGR_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: Always.

ReasonQualifier

Description: Identifier for type 1 authority events.

Identifier: MQIACF_REASON_QUALIFIER.

Data type: MQCFIN.

Values:

MQRQ_CONN_NOT_AUTHORIZED
Connection not authorized.

MQRQ_SYS_CONN_NOT_AUTHORIZED
Missing system authority.

MQRQ_CSP_NOT_AUTHORIZED
MQCSP authentication credentials not authorized.

Returned: Always.

UserIdentifier

Description: User identifier that caused the authorization check.

Identifier: MQCACF_USER_IDENTIFIER.

Data type: MQCFST.

Maximum length: MQ_USER_ID_LENGTH.

Returned: Always.

ApplType

Description: Type of application causing the event.

Identifier: MQIA_APPL_TYPE.

Data type: MQCFIN.

Returned: Always.

ApplName

Description: Name of the application causing the event.

Identifier: MQCACF_APPL_NAME.

Data type: MQCFST.

Maximum length: MQ_APPL_NAME_LENGTH.

Returned: Always.

ConnName

Description: Connection name for client connection.

Identifier: MQCACH_CONNECTION_NAME.

Data type: MQCFST.

Maximum length: MQ_CONN_NAME_LENGTH.

Returned: If the application making the MQI call that caused the event is a client attached application.

ChannelName

Description: Channel name for client connection.

Identifier: MQCACH_CHANNEL_NAME.

Data type: MQCFST.

Maximum length: MQ_CHANNEL_NAME_LENGTH.

Returned: If the application making the MQI call that caused the event is a client attached application.

CSPUserIdentifier

Description: The user identifier from the Connection Security Parameters (MQCSP) structure. If an authentication token is specified in the MQCSP structure, the value of this parameter is *TOKEN.

Identifier: MQCACF_CSP_USER_IDENTIFIER.

Data type: MQCFST.

Maximum length: MQ_CLIENT_USER_ID_LENGTH.

Returned: If the application making the MQI call that caused the event supplied a user ID or an authentication token in the MQCSP structure.

Multi *Not Authorized (type 2)*

Event name:	Not Authorized (type 2).
Reason code in MQCFH:	"2035 (07F3) (RC2035): MQRC_NOT_AUTHORIZED" on page 1169. Not authorized for access.
Event description:	On an MQOPEN or MQPUT1 call, the user is not authorized to open the object for the options specified.
Event type:	Authority.
Platforms:	All, except IBM MQ for z/OS.
Event queue:	SYSTEM.ADMIN.QMGR.EVENT.

Event data

QMgrName

Description: Name of the queue manager generating the event.

Identifier: MQCA_Q_MGR_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: Always.

ReasonQualifier

Description: Identifier for type 2 authority events.

Identifier: MQIACF_REASON_QUALIFIER.

Data type: MQCFIN.

Values:

MQRQ_OPEN_NOT_AUTHORIZED
Open not authorized.

Returned: Always.

Options

Description: Options specified on the MQOPEN call.

Identifier: MQIACF_OPEN_OPTIONS.

Data type: MQCFIN.

Returned: Always.

UserIdentifier

Description: User identifier that caused the authorization check.

Identifier: MQCACF_USER_IDENTIFIER.

Data type: MQCFST.

Maximum length: MQ_USER_ID_LENGTH.

Returned: Always.

ApplType

Description: Type of application that caused the authorization check.

Identifier: MQIA_APPL_TYPE.

Data type: MQCFIN.

Returned: Always.

ApplName

Description: Name of the application that caused the authorization check.

Identifier: MQCACF_APPL_NAME.

Data type: MQCFST.

Maximum length: MQ_APPL_NAME_LENGTH.

Returned: Always.

ObjectQMgrName

Description: Object queue manager name from object descriptor (MQOD).

Identifier: MQCACF_OBJECT_Q_MGR_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: If the **ObjectQMgrName** in the object descriptor (MQOD) when the object was opened is not the queue manager currently connected.

QName

Description: Object name from object descriptor (MQOD).

Identifier: MQCA_Q_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_NAME_LENGTH.

Returned: If the object opened is a queue object.

ProcessName

Description: Name of process object from object descriptor (MQOD).

Identifier: MQCA_PROCESS_NAME.

Data type: MQCFST.

Maximum length: MQ_PROCESS_NAME_LENGTH.

Returned: If the object opened is a process object.

TopicString

Description: Topic string being subscribed to, or opened.

Identifier: MQCA_TOPIC_STRING.

Data type: MQCFST.

Maximum length: MQ_TOPIC_STR_LENGTH.

Returned: If the object opened is a topic object.

AdminTopicNames

Description: List of topic admin objects against which authority is checked.

Identifier: MQCA_ADMIN_TOPIC_NAME.

Data type: MQCFSL.

Maximum length: MQ_TOPIC_NAME_LENGTH.

Returned: If the object opened is a topic object.

ObjectType

Description: Object type from object descriptor (MQOD).

Identifier: MQIACF_OBJECT_TYPE.

Data type: MQCFIN.

Values:

MQOT_NAMELIST
Namelist.

MQOT_PROCESS
Process.

MQOT_Q
Queue.

MQOT_Q_MGR
Queue manager.

MQOT_TOPIC
Topic.

Returned: Always.

NamelistName

Description: Object name from object descriptor (MQOD).

Identifier: MQCA_NAMELIST_NAME.

Data type: MQCFST.

Maximum length: MQ_NAMELIST_NAME_LENGTH.

Returned: If the object opened is a namelist object.

ConnName

Description: Connection name for client connection.

Identifier: MQCACH_CONNECTION_NAME.

Data type: MQCFST.

Maximum length: MQ_CONN_NAME_LENGTH.

Returned: If the application making the MQI call that caused the event is a client attached application.

ChannelName

Description: Channel name for client connection.

Identifier: MQCACH_CHANNEL_NAME.

Data type: MQCFST.

Maximum length: MQ_CHANNEL_NAME_LENGTH

Returned: If the application making the MQI call that caused the event is a client attached application.

ConnectionIdentifier

Description: The connection identifier associated with successful connections.

Identifier: MQBACF_CONNECTION_ID.

Data type: MQCFBS.

Maximum length: MQ_CONNECTION_ID_LENGTH.

Returned: Always.

ConnectionTag

Description: The connection tag associated the application.

Identifier: MQBACF_CONN_TAG.

Data type: MQCFBS.

Maximum length: MQ_CONN_TAG_LENGTH.

Returned: Always.

Multi *Not Authorized (type 3)*

Event name:	Not Authorized (type 3).
Reason code in MQCFH:	"2035 (07F3) (RC2035): MQRC_NOT_AUTHORIZED" on page 1169. Not authorized for access.
Event description:	When closing a queue using the MQCLOSE call, the user is not authorized to delete the object, which is a permanent dynamic queue, and the **Hobj** parameter specified on the MQCLOSE call is not the handle returned by the MQOPEN call that created the queue. When closing a subscription using an MQCLOSE call, the user has requested that the subscription is removed using the MQCO_REMOVE_SUB option, but the user is not the creator of the subscription or does not have *sub* authority on the topic associated with the subscription.
Event type:	Authority.
Platforms:	All, except IBM MQ for z/OS.
Event queue:	SYSTEM.ADMIN.QMGR.EVENT.

Event data

QMgrName

Description: Name of the queue manager generating the event.

Identifier: MQCA_Q_MGR_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: Always.

ReasonQualifier

Description: Identifier for type 3 authority events.

Identifier: MQIACF_REASON_QUALIFIER.

Data type: MQCFIN.

Values:

MQRQ_CLOSE_NOT_AUTHORIZED
Close not authorized.

Returned: Always.

UserIdentifier

Description: User identifier that caused the authorization check.

Identifier: MQCACF_USER_IDENTIFIER.

Data type: MQCFST.

Maximum length: MQ_USER_ID_LENGTH.

Returned: Always.

ApplType

Description: Type of application causing the authorization check.

Identifier: MQIA_APPL_TYPE.

Data type: MQCFIN.

Returned: Always.

ApplName

Description: Name of the application causing the authorization check.

Identifier: MQCACF_APPL_NAME.

Data type: MQCFST.

Maximum length: MQ_APPL_NAME_LENGTH.

Returned: Always.

QName

Description: Object name from object descriptor (MQOD).

Identifier: MQCA_Q_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_NAME_LENGTH.

Returned: If the handle being closed is a queue.

SubName

Description: Name of subscription being removed.

Identifier: MQCACF_SUB_NAME.

Data type: MQCFST.

Maximum length: MQ_SUB_NAME_LENGTH.

Returned: If the handle being closed is a subscription.

TopicString

Description: Topic string of the subscription.

Identifier: MQCA_TOPIC_STRING.

Data type: MQCFST.

Maximum length: MQ_TOPIC_STR_LENGTH.

Returned: If the handle being closed is a subscription.

AdminTopicNames

Description: List of topic administration objects against which authority was checked.

Identifier: MQCACF_ADMIN_TOPIC_NAMES.

Data type: MQCFSL.

Maximum length: MQ_TOPIC_NAME_LENGTH.

Returned: If the handle being closed is a subscription.

ConnName

Description: Connection name for client connection.

Identifier: MQCACH_CONNECTION_NAME.

Data type: MQCFST.

Maximum length: MQ_CONN_NAME_LENGTH.

Returned: If the application making the MQI call that caused the event is a client attached application.

ChannelName

Description: Channel name for client connection.

Identifier: MQCACH_CHANNEL_NAME.

Data type: MQCFST.

Maximum length: MQ_CHANNEL_NAME_LENGTH.

Returned: If the application making the MQI call that caused the event is a client attached application.

ConnectionIdentifier

Description: The connection identifier associated with successful connections.

Identifier: MQBACF_CONNECTION_ID.

Data type: MQCFBS.

Maximum length: MQ_CONNECTION_ID_LENGTH.

Returned: Always.

ConnectionTag

Description:	The connection tag associated the application
Identifier:	MQBACF_CONN_TAG.
Data type:	MQCFBS.
Maximum length:	MQ_CONN_TAG_LENGTH.
Returned:	Always.

Multi *Not Authorized (type 4)*

Event name:	Not Authorized (type 4).
Reason code in MQCFH:	"2035 (07F3) (RC2035): MQRC_NOT_AUTHORIZED" on page 1169. Not authorized for access.
Event description:	Indicates that a command has been issued from a user ID that is not authorized to access the object specified in the command.
Event type:	Authority.
Platforms:	All, except IBM MQ for z/OS.
Event queue:	SYSTEM.ADMIN.QMGR.EVENT.

Event data

QMgrName

Description:	Name of the queue manager generating the event.
Identifier:	MQCA_Q_MGR_NAME.
Data type:	MQCFST.
Maximum length:	MQ_Q_MGR_NAME_LENGTH.
Returned:	Always.

ReasonQualifier

Description:	Identifier for type 4 authority events.
Identifier:	MQIACF_REASON_QUALIFIER.
Data type:	MQCFIN.
Values:	**MQRQ_CMD_NOT_AUTHORIZED** Command not authorized.
Returned:	Always.

Command

Description:	Command identifier. See the MQCFH header structure, described in "Event message MQCFH (PCF header)" on page 125.
Identifier:	MQIACF_COMMAND.
Data type:	MQCFIN.
Returned:	Always.

UserIdentifier

Description:	User identifier that caused the authorization check.
Identifier:	MQCACF_USER_IDENTIFIER.
Data type:	MQCFST.
Maximum length:	MQ_USER_ID_LENGTH.
Returned:	Always.

Multi *Not Authorized (type 5)*

Event name:	Not Authorized (type 5).
Reason code in MQCFH:	"2035 (07F3) (RC2035): MQRC_NOT_AUTHORIZED" on page 1169. Not authorized for access.
Event description:	On an MQSUB call, the user is not authorized to subscribe to the specified topic.
Event type:	Authority.
Platforms:	All, except IBM MQ for z/OS.
Event queue:	SYSTEM.ADMIN.QMGR.EVENT.

Event data

QMgrName

Description:	Name of the queue manager generating the event.
Identifier:	MQCA_Q_MGR_NAME.
Data type:	MQCFST.
Maximum length:	MQ_Q_MGR_NAME_LENGTH.
Returned:	Always.

ReasonQualifier

Description:	Identifier for type 5 authority events.
Identifier:	MQIACF_REASON_QUALIFIER.
Data type:	MQCFIN.
Values:	**MQRQ_SUB_NOT_AUTHORIZED** Subscribe not authorized.
Returned:	Always.

Options

Description:	Options specified on the MQSUB call.
Identifier:	MQIACF_SUB_OPTIONS.
Data type:	MQCFIN.
Returned:	Always.

UserIdentifier

Description:	User identifier that caused the authorization check.
Identifier:	MQCACF_USER_IDENTIFIER.
Data type:	MQCFST.
Maximum length:	MQ_USER_ID_LENGTH.
Returned:	Always.

ApplType

Description:	Type of application that caused the authorization check.
Identifier:	MQIA_APPL_TYPE.
Data type:	MQCFIN.
Returned:	Always.

ApplName

Description:	Name of the application that caused the authorization check.
Identifier:	MQCACF_APPL_NAME.
Data type:	MQCFST.
Maximum length:	MQ_APPL_NAME_LENGTH.
Returned:	Always.

TopicString

Description:	Topic string being opened or subscribed to.
Identifier:	MQCA_TOPIC_STRING.
Data type:	MQCFST.
Maximum length:	MQ_TOPIC_STR_LENGTH.
Returned:	Always.

AdminTopicNames

Description:	List of topic administration objects against which authority is checked.
Identifier:	MQCACF_ADMIN_TOPIC_NAMES.
Data type:	MQCFSL.
Maximum length for each member of the string list:	MQ_TOPIC_NAME_LENGTH.
Returned:	Always.

ConnName

Description:	Connection name for client connection.
Identifier:	MQCACH_CONNECTION_NAME.
Data type:	MQCFST.
Maximum length:	MQ_CONN_NAME_LENGTH.

Returned: If the application making the MQI call that caused the event is a client attached application.

ChannelName

Description: Channel name for client connection.

Identifier: MQCACH_CHANNEL_NAME.

Data type: MQCFST.

Maximum length: MQ_CHANNEL_NAME_LENGTH.

Returned: If the application making the MQI call that caused the event is a client attached application.

ConnectionIdentifier

Description: The connection identifier associated with successful connections.

Identifier: MQBACF_CONNECTION_ID.

Data type: MQCFBS.

Maximum length: MQ_CONNECTION_ID_LENGTH.

Returned: Always.

ConnectionTag

Description: The connection tag associated the application.

Identifier: MQBACF_CONN_TAG.

Data type: MQCFBS.

Maximum length: MQ_CONN_TAG_LENGTH.

Returned: Always.

Multi *Not Authorized (type 6)*

Event name:	Not Authorized (type 6).
Reason code in MQCFH:	"2035 (07F3) (RC2035): MQRC_NOT_AUTHORIZED" on page 1169. Not authorized for access.
Event description:	On an MQSUB call, the user is not authorized to use the destination queue with the required level of access. This event is only returned for subscriptions using non-managed destination queues. When creating, altering, or resuming a subscription, and a handle to the destination queue is supplied on the request, the user does not have PUT authority on the destination queue provided. When resuming or alerting a subscription and the handle to the destination queue is to be returned on the MQSUB call, and the user does not have PUT, GET and BROWSE authority on the destination queue.
Event type:	Authority.
Platforms:	All, except IBM MQ for z/OS.
Event queue:	SYSTEM.ADMIN.QMGR.EVENT.

Event data

QMgrName

Description: Name of the queue manager generating the event.

Identifier: MQCA_Q_MGR_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: Always.

ReasonQualifier

Description: Identifier for type 6 authority events.

Identifier: MQIACF_REASON_QUALIFIER.

Data type: MQCFIN.

Values:

MQRQ_SUB_DEST_NOT_AUTHORIZED
Subscription destination queue usage not authorized.

Returned: Always.

Options

Description: Options specified on the MQSUB call.

Identifier: MQIACF_SUB_OPTIONS.

Data type: MQCFIN.

Returned: Always.

UserIdentifier

Description: User identifier that caused the authorization check.

Identifier: MQCACF_USER_IDENTIFIER.

Data type: MQCFST.

Maximum length: MQ_USER_ID_LENGTH.

Returned: Always.

ApplType

Description: Type of application that caused the authorization check.

Identifier: MQIA_APPL_TYPE.

Data type: MQCFIN.

Returned: Always.

ApplName

Description: Name of the application that caused the authorization check.

Identifier: MQCACF_APPL_NAME.

Data type: MQCFST.

Maximum length: MQ_APPL_NAME_LENGTH.

Returned: Always.

TopicString

Description: Topic string being subscribed to.

Identifier: MQCA_TOPIC_STRING.

Data type: MQCFST.

Maximum length: MQ_TOPIC_STR_LENGTH.

Returned: Always.

DestQMgrName

Description: Hosting queue manager name of the subscription's destination queue.

Identifier: MQCACF_OBJECT_Q_MGR_NAME

Data type: MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: If the queue manager hosting the destination queue is not the queue manager to which the application is currently connected.

DestQName

Description: The name of the destination queue of the subscription..

Identifier: MQCA_Q_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_NAME_LENGTH.

Returned: Always.

DestOpenOptions

Description: The open options requested for the destination queue.

Identifier: MQIACF_OPEN_OPTIONS.

Data type: MQCFIN.

Returned: Always.

ConnName

Description: Connection name for client connection.

Identifier: MQCACH_CONNECTION_NAME.

Data type: MQCFST.

Maximum length: MQ_CONN_NAME_LENGTH.

Returned: If the application making the MQI call that caused the event is a client attached application.

ChannelName

Description: Channel name for client connection.

Identifier: MQCACH_CHANNEL_NAME.

Data type: MQCFST.

Maximum length: MQ_CHANNEL_NAME_LENGTH.

Returned: If the application making the MQI call that caused the event is a client attached application.

V 10.0.0 ConnectionIdentifier

Description: The connection identifier associated with successful connections.

Identifier: MQBACF_CONNECTION_ID.

Data type: MQCFBS.

Maximum length: MQ_CONNECTION_ID_LENGTH.

Returned: Always.

V 10.0.0 ConnectionTag

Description: The connection tag associated the application.

Identifier: MQBACF_CONN_TAG.

Data type: MQCFBS.

Maximum length: MQ_CONN_TAG_LENGTH.

Returned: Always.

Put Inhibited

Event name:	Put Inhibited.
Reason code in MQCFH:	“2051 (0803) (RC2051): MQRC_PUT_INHIBITED” on page 1177. Put calls inhibited for the queue or topic.
Event description:	MQPUT and MQPUT1 calls are currently inhibited for the queue or topic (see the **InhibitPut** queue attribute in InhibitPut (MQLONG) or the **InhibitPublications** topic attribute in “Topic attributes” on page 114 for the queue to which this queue resolves.
Event type:	Inhibit.
Platforms:	All.
Event queue:	SYSTEM.ADMIN.QMGR.EVENT.

Event data

QMgrName

Description: Name of the queue manager generating the event.

Identifier: MQCA_Q_MGR_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: Always.

QName

Description: Queue name from object descriptor (MQOD).

Identifier: MQCA_Q_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_NAME_LENGTH.

Returned: If the object opened is a queue object

ApplType

Description: Type of application that issued the put.

Identifier: MQIA_APPL_TYPE.

Data type: MQCFIN.

Returned: Always.

ApplName

Description: Name of the application that issued the put.

Identifier: MQCACF_APPL_NAME.

Data type: MQCFST.

Maximum length: MQ_APPL_NAME_LENGTH.

Returned: Always.

ObjectQMgrName

Description: Name of queue manager from object descriptor (MQOD).

Identifier: MQCACF_OBJECT_Q_MGR_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: Only if this parameter has a value different from *QMgrName*. This occurs when the *ObjectQMgrName* field in the object descriptor provided by the application on the MQOPEN or MQPUT1 call is neither blank nor the name of the application's local queue manager. However, it can also occur when *ObjectQMgrName* in the object descriptor is blank, but a name service provides a queue manager name that is not the name of the application's local queue manager.

TopicString

Description: Topic String being opened

Identifier: MQCA_TOPIC_STRING

Data type: MQCFST.

Maximum length: MQ_TOPIC_STR_LENGTH.

Returned: If the object opened is a topic.

ConnName

Description: Connection name for client connection.

Identifier: MQCACH_CONNECTION_NAME.

Data type: MQCFST.

Maximum length: MQ_CONN_NAME_LENGTH.

Returned: If the application making the MQI call that caused the event is a client attached application.

ChannelName

Description:	Channel name for client connection.
Identifier:	MQCACH_CHANNEL_NAME.
Data type:	MQCFST.
Maximum length:	MQ_CHANNEL_NAME_LENGTH
Returned:	If the application making the MQI call that caused the event is a client attached application.

Related reference

IBM i InhibitPut (10-digit signed integer)
Inquire Queue (Response)
Inquire Topic (Response)
Inquire Topic Status (Response)
Change, Copy, and Create Topic

Queue Depth High

Event name:	Queue Depth High.
Reason code in MQCFH:	"2224 (08B0) (RC2224): MQRC_Q_DEPTH_HIGH" on page 1236. Queue depth high limit reached or exceeded.
Event description:	An MQPUT or MQPUT1 call has caused the queue depth to be incremented to or above the limit specified in the **QDepthHighLimit** attribute.
Event type:	Performance.
Platforms:	All.
Event queue:	SYSTEM.ADMIN.PERFM.EVENT.

Note:

1. IBM MQ for z/OS supports queue depth events on shared queues. You might receive a NULL event message for a shared queue if a queue manager has performed no activity on that shared queue.
2. For shared queues, the correlation identifier, *CorrelId* in the message descriptor (MQMD) is set. See "Event message MQMD (message descriptor)" on page 121 for more information.

Event data

QMgrName

Description:	Name of the queue manager generating the event.
Identifier:	MQCA_Q_MGR_NAME.
Data type:	MQCFST.
Maximum length:	MQ_Q_MGR_NAME_LENGTH.
Returned:	Always.

QName

Description:	Name of the queue on which the limit has been reached.
Identifier:	MQCA_BASE_Q_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_NAME_LENGTH.

Returned: Always.

TimeSinceReset

Description: Time, in seconds, since the statistics were last reset. The value recorded by this timer is also used as the *interval time* in queue service interval events.

Identifier: MQIA_TIME_SINCE_RESET.

Data type: MQCFIN.

Returned: Always.

HighQDepth

Description: Maximum number of messages on the queue since the queue statistics were last reset.

Identifier: MQIA_HIGH_Q_DEPTH.

Data type: MQCFIN.

Returned: Always.

MsgEnqCount

Description: Number of messages enqueued. This is the number of messages put on the queue since the queue statistics were last reset.

Identifier: MQIA_MSG_ENQ_COUNT.

Data type: MQCFIN.

Returned: Always.

MsgDeqCount

Description: Number of messages removed from the queue since the queue statistics were last reset.

Identifier: MQIA_MSG_DEQ_COUNT.

Data type: MQCFIN.

Returned: Always.

Queue Depth Low

Event name:	Queue Depth Low.
Reason code in MQCFH:	“2225 (08B1) (RC2225): MQRC_Q_DEPTH_LOW” on page 1237. Queue depth low limit reached or exceeded.
Event description:	A get operation has caused the queue depth to be decremented to or below the limit specified in the **QDepthLowLimit** attribute.
Event type:	Performance.
Platforms:	All.
Event queue:	SYSTEM.ADMIN.PERFM.EVENT.

Note:

1. IBM MQ for z/OS supports queue depth events on shared queues. You might receive a NULL event message for a shared queue if a queue manager has performed no activity on that shared queue.
2. For shared queues, the correlation identifier, *CorrelId* in the message descriptor (MQMD) is set. See "Event message MQMD (message descriptor)" on page 121 for more information.

Event data

QMgrName

Description: Name of the queue manager generating the event.

Identifier: MQCA_Q_MGR_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: Always.

QName

Description: Name of the queue on which the limit has been reached.

Identifier: MQCA_BASE_Q_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_NAME_LENGTH.

Returned: Always.

TimeSinceReset

Description: Time, in seconds, since the statistics were last reset. The value recorded by this timer is also used as the *interval time* in queue service interval events.

Identifier: MQIA_TIME_SINCE_RESET.

Data type: MQCFIN.

Returned: Always.

HighQDepth

Description: Maximum number of messages on the queue since the queue statistics were last reset.

Identifier: MQIA_HIGH_Q_DEPTH.

Data type: MQCFIN.

Returned: Always.

MsgEnqCount

Description: Number of messages enqueued. This is the number of messages put on the queue since the queue statistics were last reset.

Identifier: MQIA_MSG_ENQ_COUNT.

Data type: MQCFIN.

Returned: Always.

MsgDeqCount

Description:	Number of messages removed from the queue since the queue statistics were last reset.
Identifier:	MQIA_MSG_DEQ_COUNT.
Data type:	MQCFIN.
Returned:	Always.

Queue Full

Event name:	Queue Full.
Reason code in MQCFH:	"2053 (0805) (RC2053): MQRC_Q_FULL" on page 1178. Queue already contains maximum number of messages.
Event description:	On an MQPUT or MQPUT1 call, the call failed because the queue is full. That is, it already contains the maximum number of messages possible (see the *MaxQDepth* local-queue attribute
Event type:	Performance.
Platforms:	All.
Event queue:	SYSTEM.ADMIN.PERFM.EVENT.

Note:

1. IBM MQ for z/OS supports queue depth events on shared queues. You might receive a NULL event message for a shared queue if a queue manager has performed no activity on that shared queue.
2. For shared queues, the correlation identifier, *CorrelId* in the message descriptor (MQMD) is set. See "Event message MQMD (message descriptor)" on page 121 for more information.

Event data

QMgrName

Description:	Name of the queue manager generating the event.
Identifier:	MQCA_Q_MGR_NAME.
Data type:	MQCFST.
Maximum length:	MQ_Q_MGR_NAME_LENGTH.
Returned:	Always.

QName

Description:	Name of the queue on which the put was rejected.
Identifier:	MQCA_BASE_Q_NAME.
Data type:	MQCFST.
Maximum length:	MQ_Q_NAME_LENGTH.
Returned:	Always.

TimeSinceReset

Description:	Time, in seconds, since the statistics were last reset.
Identifier:	MQIA_TIME_SINCE_RESET.

Data type: MQCFIN.

Returned: Always.

HighQDepth

Description: Maximum number of messages on a queue.

Identifier: MQIA_HIGH_Q_DEPTH.

Data type: MQCFIN.

Returned: Always.

MsgEnqCount

Description: Number of messages enqueued. This is the number of messages put on the queue since the queue statistics were last reset.

Identifier: MQIA_MSG_ENQ_COUNT.

Data type: MQCFIN.

Returned: Always.

MsgDeqCount

Description: Number of messages removed from the queue since the queue statistics were last reset.

Identifier: MQIA_MSG_DEQ_COUNT.

Data type: MQCFIN.

Returned: Always.

Queue Manager Active

Event name:	Queue Manager Active.
Reason code in MQCFH:	"2222 (08AE) (RC2222): MQRC_Q_MGR_ACTIVE" on page 1236. Queue manager active.
Event description:	This condition is detected when a queue manager becomes active.
Event type:	Start And Stop.
Platforms:	All, except the first start of an IBM MQ for z/OS queue manager. In this case it is produced only on subsequent restarts. The *ReasonQualifier* and *HostName* fields apply only to those platforms that support multi-instance availability; that is not z/OS
Event queue:	SYSTEM.ADMIN.QMGR.EVENT.

Event data

QMgrName

Description: Name of the queue manager generating the event.

Identifier: MQCA_Q_MGR_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: Always.

ReasonQualifier

Description: Identifier of causes for this reason code. This specifies the type of start that is happening.

Identifier: MQIACF_REASON_QUALIFIER.

Data type: MQCFIN.

Values:

MQRQ_FAILOVER_PERMITTED

Queue manager has started normally and allows a standby instance.

MQRQ_FAILOVER_NOT_PERMITTED

Queue manager has started normally but does not allow a standby instance.

MQRQ_STANDBY_ACTIVATED
Queue manager has moved out of standby mode into active mode.

MQRQ_REPLICA_ACTIVATED
A Native HA queue manager replica instance has moved into active mode.

Returned: Always.

HostName

Description: The host name of the machine on which the queue manager is running.

Identifier: MQCACF_HOST_NAME.

Data type: MQCFST.

Returned: Always.

Queue Manager Not Active

Event name:	Queue Manager Not Active.
Reason code in MQCFH:	"2223 (08AF) (RC2223): MQRC_Q_MGR_NOT_ACTIVE" on page 1236. Queue manager unavailable.
Event description:	This condition is detected when a queue manager is requested to stop or quiesce.
Event type:	Start And Stop.
Platforms:	All, except IBM MQ for z/OS.
Event queue:	SYSTEM.ADMIN.QMGR.EVENT.

Event data

QMgrName

Description: Name of the queue manager generating the event.

Identifier: MQCA_Q_MGR_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: Always.

ReasonQualifier

Description: Identifier of causes of this reason code. This specifies the type of stop that was requested.

Identifier: MQIACF_REASON_QUALIFIER.

Data type: MQCFIN.

Values:

MQRQ_Q_MGR_STOPPING
Queue manager stopping.

MQRQ_Q_MGR_QUIESCING
Queue manager quiescing.

Returned: Always.

Queue Service Interval High

Event name:	Queue Service Interval High.
Reason code in MQCFH:	“2226 (08B2) (RC2226): MQRC_Q_SERVICE_INTERVAL_HIGH” on page 1237. Queue service interval high.
Event description:	No successful get operations or MQPUT calls have been detected within an interval greater than the limit specified in the **QServiceInterval** attribute.
Event type:	Performance.
Platforms:	All.
Event queue:	SYSTEM.ADMIN.PERFM.EVENT.

Note: IBM MQ for z/OS does not support service interval events on shared queues.

Event data

QMgrName

Description: Name of the queue manager generating the event.

Identifier: MQCA_Q_MGR_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: Always.

QName

Description: Name of the queue specified on the command that caused this queue service interval event to be generated.

Identifier: MQCA_BASE_Q_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_NAME_LENGTH.

Returned: Always.

TimeSinceReset

Description: Time, in seconds, since the statistics were last reset. For a service interval high event, this value is greater than the service interval.

Identifier: MQIA_TIME_SINCE_RESET.

Data type: MQCFIN.

Returned: Always.

HighQDepth

Description: Maximum number of messages on the queue since the queue statistics were last reset.

Identifier: MQIA_HIGH_Q_DEPTH.

Data type: MQCFIN.

Returned: Always.

MsgEnqCount

Description: Number of messages enqueued. This is the number of messages put on the queue since the queue statistics were last reset.

Identifier: MQIA_MSG_ENQ_COUNT.

Data type: MQCFIN.

Returned: Always.

MsgDeqCount

Description: Number of messages removed from the queue since the queue statistics were last reset.

Identifier: MQIA_MSG_DEQ_COUNT.

Data type: MQCFIN.

Returned: Always.

Queue Service Interval OK

Event name:	Queue Service Interval OK.
Reason code in MQCFH:	"2227 (08B3) (RC2227): MQRC_Q_SERVICE_INTERVAL_OK" on page 1237. Queue service interval OK.
Event description:	A successful get operation has been detected within an interval less than or equal to the limit specified in the **QServiceInterval** attribute.
Event type:	Performance.
Platforms:	All.
Event queue:	SYSTEM.ADMIN.PERFM.EVENT.

Note: IBM MQ for z/OS does not support service interval events on shared queues.

Event data

QMgrName

Description: Name of the queue manager generating the event.

Identifier: MQCA_Q_MGR_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: Always.

QName

Description: Queue name specified on the command that caused this queue service interval event to be generated.

Identifier: MQCA_BASE_Q_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_NAME_LENGTH.

Returned: Always.

TimeSinceReset

Description: Time, in seconds, since the statistics were last reset.

Identifier: MQIA_TIME_SINCE_RESET.

Data type: MQCFIN.

Returned: Always.

HighQDepth

Description: Maximum number of messages on the queue since the queue statistics were last reset.

Identifier: MQIA_HIGH_Q_DEPTH.

Data type: MQCFIN.

Returned: Always.

MsgEnqCount

Description: Number of messages enqueued. This is the number of messages put on the queue since the queue statistics were last reset.

Identifier: MQIA_MSG_ENQ_COUNT.

Data type: MQCFIN.

Returned: Always.

MsgDeqCount

Description: Number of messages removed from the queue since the queue statistics were last reset.

Identifier: MQIA_MSG_DEQ_COUNT.

Data type: MQCFIN.

Returned: Always.

Queue Type Error

Event name:	Queue Type Error.
Reason code in MQCFH:	"2057 (0809) (RC2057): MQRC_Q_TYPE_ERROR" on page 1179. Queue type not valid.
Event description:	On an MQOPEN call, the *ObjectQMgrName* field in the object descriptor specifies the name of a local definition of a remote queue (in order to specify a queue manager alias). In that local definition the **RemoteQMgrName** attribute is the name of the local queue manager. However, the *ObjectName* field specifies the name of a model queue on the local queue manager, which is not allowed. See the Queue manager events for more information.
Event type:	Remote.
Platforms:	All.
Event queue:	SYSTEM.ADMIN.QMGR.EVENT.

Event data

QMgrName

Description:	Name of the queue manager generating the event.
Identifier:	MQCA_Q_MGR_NAME.
Data type:	MQCFST.
Maximum length:	MQ_Q_MGR_NAME_LENGTH.
Returned:	Always.

QName

Description:	Queue name from object descriptor (MQOD).
Identifier:	MQCA_Q_NAME.
Data type:	MQCFST.
Maximum length:	MQ_Q_NAME_LENGTH.
Returned:	Always.

ApplType

Description:	Type of application making the MQI call that caused the event.
Identifier:	MQIA_APPL_TYPE.
Data type:	MQCFIN.
Returned:	Always.

ApplName

Description:	Name of the application making the MQI call that caused the event.
Identifier:	MQCACF_APPL_NAME.
Data type:	MQCFST.
Maximum length:	MQ_APPL_NAME_LENGTH.
Returned:	Always.

ObjectQMgrName

Description:	Name of the object queue manager.
Identifier:	MQCACF_OBJECT_Q_MGR_NAME.
Data type:	MQCFST.
Maximum length:	MQ_Q_MGR_NAME_LENGTH.
Returned:	Always.

ConnName

Description:	Connection name for client connection.
Identifier:	MQCACH_CONNECTION_NAME.
Data type:	MQCFST.
Maximum length:	MQ_CONN_NAME_LENGTH.
Returned:	If the application making the MQI call that caused the event is a client attached application.

ChannelName

Description:	Channel name for client connection.
Identifier:	MQCACH_CHANNEL_NAME.
Data type:	MQCFST.
Maximum length:	MQ_CHANNEL_NAME_LENGTH
Returned:	If the application making the MQI call that caused the event is a client attached application.

Refresh Authority Record

Event name:	Refresh Authority Record
Reason code in MQCFH:	"2370 (0942) (RC2370): MQRC_CONFIG_REFRESH_OBJECT" on page 1286. Refresh queue manager configuration - authority records.
Event description:	A REFRESH QMGR command specifying **TYPE**(*CONFIGEV*) was issued.
Event type:	Configuration
Platforms:	All except z/OS.
Event queue:	SYSTEM.ADMIN.CONFIG.EVENT.

Note that the REFRESH QMGR command can produce many configuration events; one event is generated for each authority record that is selected by the command.

Event data

EventQMgr

Description:	The queue manager where the command or call was entered. That is, the queue manager where the command is processed and that generates the event is in the MQMD of the event message.
Identifier:	MQCACF_EVENT_Q_MGR
Data type:	MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: Always.

EventUserId

Description: The user ID that issued the command or call that generated the event.

This is the same user ID that is used to check the authority to issue the command or call; for commands received from a queue, this is also the user identifier (**UserIdentifier**) from the message descriptor of the command message.

.

Identifier: MQCACF_EVENT_USER_ID

Data type: MQCFST.

Maximum length: MQ_USER_ID_LENGTH.

Returned: Always.

EventOrigin

Description: The origin of the action causing the event.

Identifier: MQIACF_EVENT_ORIGIN

Data type: MQCFIN.

Values:

MQEVO_CONSOLE
Console command (runmqsc or setmqaut)

MQEVO_INTERNAL
Directly by queue manager

MQEVO_MSG
Command message on SYSTEM.ADMIN.COMMAND.QUEUE

Returned: Always

EventAccountingToken

Description: For commands received as a message (MQEVO_MSG), the accounting token (**AccountingToken**) from the message descriptor of the command message.

Identifier: MQBACF_EVENT_ACCOUNTING_TOKEN

Data type: MQCFBS

Maximum length: MQ_ACCOUNTING_TOKEN_LENGTH

Returned: Only if **EventOrigin** is MQEVO_MSG.

EventApplIdentity

Description: For commands received as a message (MQEVO_MSG), application identity data (**ApplIdentityData**) from the message descriptor of the command message.

Identifier: MQMQCACF_EVENT_APPL_IDENTITY

Data type: MQCFST

Maximum length: MQ_APPL_IDENTITY_DATA_LENGTH

Returned: Only if **EventOrigin** is MQEVO_MSG.

EventApplType

Description: For commands received as a message (MQEVO_MSG), the type of application (**PutApplType**) from the message descriptor of the command message.

Identifier: MQIACF_EVENT_APPL_TYPE

Data type: MQCFIN

Values:

Returned: Only if **EventOrigin** is MQEVO_MSG.

EventApplName

Description: For commands received as a message (MQEVO_MSG), the name of the application (**PutApplName**) from the message descriptor of the command message.

Identifier: MQCACF_EVENT_APPL_NAME

Data type: MQCFST

Maximum length: MQ_APPL_NAME_LENGTH

Returned: Only if **EventOrigin** is MQEVO_MSG.

EventApplOrigin

Description: For commands received as a message (MQEVO_MSG), the application origin data (**ApplOriginData**) from the message descriptor of the command message.

Identifier: MQCACF_EVENT_APPL_ORIGIN

Data type: MQCFST

Maximum length: MQ_APPL_ORIGIN_DATA_LENGTH

Returned: Only if **EventOrigin** is MQEVO_MSG.

ObjectType

Description: Object type

Identifier: MQIACF_OBJECT_TYPE

Data type: MQCFIN

Values: MQOT_AUTH_REC

Returned: Always

ProfileName

Description: Object or generic profile name

Identifier: MQCACF_AUTH_PROFILE_NAME

Data type: MQCFST

Maximum length: MQ_AUTH_PROFILE_NAME_LENGTH

Returned: Always

Object attributes

A parameter structure is returned for each attribute of the object. The attributes returned depend on the object type. For more information see “Authority Record attributes” on page 119.

Refresh object

Event name:	Refresh object.
Reason code in MQCFH:	"2370 (0942) (RC2370): MQRC_CONFIG_REFRESH_OBJECT" on page 1286. Refresh queue manager configuration.
Event description:	A REFRESH QMGR command specifying TYPE (CONFIGEV) was issued.
Event type:	Configuration.
Platforms:	All.
Event queue:	SYSTEM.ADMIN.CONFIG.EVENT.

Note: The REFRESH QMGR command can produce many configuration events; one event is generated for each object that is selected by the command.

Event data

EventUserId

Description: The user ID that issued the command or call that generated the event. (This is the same user ID that is used to check the authority to issue the command or call; for commands received from a queue, this is also the user identifier (UserIdentifier) from the MQMD of the command message).

Identifier: MQCACF_EVENT_USER_ID.

Data type: MQCFST.

Maximum length: MQ_USER_ID_LENGTH.

Returned: Always.

EventOrigin

Description: The origin of the action causing the event.

Identifier: MQIACF_EVENT_ORIGIN.

Data type: MQCFIN.

Values:

MQEVO_CONSOLE
Console command.

MQEVO_INIT
Initialization input data set command.

MQEVO_INTERNAL
Directly by queue manager.

MQEVO_MSG
Command message on SYSTEM.COMMAND.INPUT.

MQEVO_OTHER
None of the above.

Returned: Always.

EventQMgr

Description: The queue manager where the command or call was entered. (The queue manager where the command is executed and that generates the event is in the MQMD of the event message).

Identifier: MQCACF_EVENT_Q_MGR.

Data type: MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: Always.

EventAccountingToken

Description: For commands received as a message (MQEVO_MSG), the accounting token (AccountingToken) from the MQMD of the command message.

Identifier: MQBACF_EVENT_ACCOUNTING_TOKEN.

Data type: MQCFBS.

Maximum length: MQ_ACCOUNTING_TOKEN_LENGTH.

Returned: Only if EventOrigin is MQEVO_MSG.

EventApplIdentity

Description: For commands received as a message (MQEVO_MSG), application identity data (ApplIdentityData) from the MQMD of the command message.

Identifier: MQCACF_EVENT_APPL_IDENTITY.

Data type: MQCFST.

Maximum length: MQ_APPL_IDENTITY_DATA_LENGTH.

Returned: Only if EventOrigin is MQEVO_MSG.

EventApplType

Description: For commands received as a message (MQEVO_MSG), the type of application (PutApplType) from the MQMD of the command message.

Identifier: MQIACF_EVENT_APPL_TYPE.

Data type: MQCFIN.

Returned: Only if EventOrigin is MQEVO_MSG.

EventApplName

Description: For commands received as a message (MQEVO_MSG), the name of the application (PutApplName) from the MQMD of the command message.

Identifier: MQCACF_EVENT_APPL_NAME.

Data type: MQCFST.

Maximum length: MQ_APPL_NAME_LENGTH.

Returned: Only if EventOrigin is MQEVO_MSG.

EventApplOrigin

Description: For commands received as a message (MQEVO_MSG), the application origin data (ApplOriginData) from the MQMD of the command message.

Identifier: MQCACF_EVENT_APPL_ORIGIN.

Data type: MQCFST.

Maximum length: MQ_APPL_ORIGIN_DATA_LENGTH.

Returned: Only if EventOrigin is MQEVO_MSG.

ObjectType

Description: Object type:

Identifier: MQIACF_OBJECT_TYPE.

Data type: MQCFIN.

Values:

MQOT_CHANNEL
Channel.

MQOT_CHLAUTH
Channel authentication record.

MQOT_NAMELIST
Namelist.

MQOT_NONE
No object.

MQOT_PROCESS
Process.

MQOT_Q
Queue.

MQOT_Q_MGR
Queue manager.

MQOT_STORAGE_CLASS
Storage class.

MQOT_AUTH_INFO
Authentication information.

MQOT_CF_STRUC
CF structure.

MQOT_TOPIC
Topic.

MQOT_COMM_INFO
Communication information.

MQOT_LISTENER
Channel Listener.

Returned: Always.

ObjectName

Description: Object name:

Identifier : Identifier will be according to object type.

- MQCACH_CHANNEL_NAME
- MQCA_NAMELIST_NAME
- MQCA_PROCESS_NAME
- MQCA_Q_NAME
- MQCA_Q_MGR_NAME
- MQCA_STORAGE_CLASS
- MQCA_AUTH_INFO_NAME
- MQCA_CF_STRUC_NAME
- MQCA_TOPIC_NAME
- MQCA_COMM_INFO_NAME
- MQCACH_LISTENER_NAME

Note: MQCACH_CHANNEL_NAME can also be used for channel authentication.

Data type: MQCFST.

Maximum length: MQ_OBJECT_NAME_LENGTH.

Returned: Always

Disposition

Description: Object disposition:

Identifier: MQIA_QSG_DISP.

Data type: MQCFIN.

Values:

MQQSGD_Q_MGR
Object resides on page set of queue manager.

MQQSGD_SHARED
Object resides in shared repository and messages are shared in coupling facility.

MQQSGD_GROUP
Object resides in shared repository.

MQQSGD_COPY
Object resides on page set of queue manager and is a local copy of a GROUP object.

Returned: Always, except for queue manager and CF structure objects.

Object attributes

A parameter structure is returned for each attribute of the object. The attributes returned depend on the object type. For more information see "Object attributes for event data" on page 74.

Remote Queue Name Error

Event name:	Remote Queue Name Error.
Reason code in MQCFH:	"2184 (0888) (RC2184): MQRC_REMOTE_Q_NAME_ERROR" on page 1223. Remote queue name not valid.

Event description:	On an MQOPEN or MQPUT1 call one of the following occurs: • A local definition of a remote queue (or an alias to one) was specified, but the **RemoteQName** attribute in the remote queue definition is blank. Note that this error occurs even if the *XmitQName* in the definition is not blank. • The *ObjectQMgrName* field in the object descriptor is not blank and not the name of the local queue manager, but the *ObjectName* field is blank.
Event type:	Remote.
Platforms:	All.
Event queue:	SYSTEM.ADMIN.QMGR.EVENT.

Event data

QMgrName

Description:	Name of the queue manager generating the event.
Identifier:	MQCA_Q_MGR_NAME.
Data type:	MQCFST.
Maximum length:	MQ_Q_MGR_NAME_LENGTH.
Returned:	Always.

QName

Description:	Queue name from object descriptor (MQOD).
Identifier:	MQCA_Q_NAME.
Data type:	MQCFST.
Maximum length:	MQ_Q_NAME_LENGTH.
Returned:	Always.

ApplType

Description:	Type of application making the MQI call that caused the event.
Identifier:	MQIA_APPL_TYPE.
Data type:	MQCFIN.
Returned:	Always.

ApplName

Description:	Name of the application making the MQI call that caused the event.
Identifier:	MQCACF_APPL_NAME.
Data type:	MQCFST.
Maximum length:	MQ_APPL_NAME_LENGTH.
Returned:	Always.

ObjectQMgrName

Description:	Name of the object queue manager.
Identifier:	MQCACF_OBJECT_Q_MGR_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: If the *ObjectName* in the object descriptor (MQOD), when the object was opened, is not the queue manager currently connected.

ConnName

Description: Connection name for client connection.

Identifier: MQCACH_CONNECTION_NAME.

Data type: MQCFST.

Maximum length: MQ_CONN_NAME_LENGTH.

Returned: If the application making the MQI call that caused the event is a client attached application.

ChannelName

Description: Channel name for client connection.

Identifier: MQCACH_CHANNEL_NAME.

Data type: MQCFST.

Maximum length: MQ_CHANNEL_NAME_LENGTH

Returned: If the application making the MQI call that caused the event is a client attached application.

Transmission Queue Type Error

Event name:	Transmission Queue Type Error.
Reason code in MQCFH:	"2091 (082B) (RC2091): MQRC_XMIT_Q_TYPE_ERROR" on page 1192. Transmission queue not local.
Event description:	On an MQOPEN or MQPUT1 call, a message is to be sent to a remote queue manager. The *ObjectName* or *ObjectQMgrName* field in the object descriptor specifies the name of a local definition of a remote queue but one of the following applies to the **XmitQName** attribute of the definition. Either: • *XmitQName* is not blank, but specifies a queue that is not a local queue, or • *XmitQName* is blank, but *RemoteQMgrName* specifies a queue that is not a local queue This also occurs if the queue name is resolved through a cell directory, and the remote queue manager name obtained from the cell directory is the name of a queue, but this is not a local queue.
Event type:	Remote.
Platforms:	All.
Event queue:	SYSTEM.ADMIN.QMGR.EVENT.

Event data

QMgrName

Description: Name of the queue manager generating the event.

Identifier: MQCA_Q_MGR_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: Always.

QName

Description: Queue name from object descriptor (MQOD).

Identifier: MQCA_Q_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_NAME_LENGTH.

Returned: Always.

XmitQName

Description: Transmission queue name.

Identifier: MQCA_XMIT_Q_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_NAME_LENGTH.

Returned: Always.

QType

Description: Type of transmission queue.

Identifier: MQIA_Q_TYPE.

Data type: MQCFIN.

Values:

MQQT_ALIAS
Alias queue definition.

MQQT_REMOTE
Local definition of a remote queue.

Returned: Always.

ApplType

Description: Type of application making the MQI call that caused the event.

Identifier: MQIA_APPL_TYPE.

Data type: MQCFIN.

Returned: Always.

ApplName

Description: Name of the application making the MQI call that caused the event.

Identifier: MQCACF_APPL_NAME.

Data type: MQCFST.

Maximum length: MQ_APPL_NAME_LENGTH.

Returned: Always.

ObjectQMgrName

Description:	Name of the object queue manager.
Identifier:	MQCACF_OBJECT_Q_MGR_NAME.
Data type:	MQCFST.
Maximum length:	MQ_Q_MGR_NAME_LENGTH.
Returned:	If the *ObjectName* in the object descriptor (MQOD), when the object was opened, is not the queue manager currently connected.

ConnName

Description:	Connection name for client connection.
Identifier:	MQCACH_CONNECTION_NAME.
Data type:	MQCFST.
Maximum length:	MQ_CONN_NAME_LENGTH.
Returned:	If the application making the MQI call that caused the event is a client attached application.

ChannelName

Description:	Channel name for client connection.
Identifier:	MQCACH_CHANNEL_NAME.
Data type:	MQCFST.
Maximum length:	MQ_CHANNEL_NAME_LENGTH
Returned:	If the application making the MQI call that caused the event is a client attached application.

Transmission Queue Usage Error

Event name:	Transmission Queue Usage Error.
Reason code in MQCFH:	"2092 (082C) (RC2092): MQRC_XMIT_Q_USAGE_ERROR" on page 1193. Transmission queue with wrong usage.
Event description:	On an MQOPEN or MQPUT1 call, a message is to be sent to a remote queue manager, but one of the following occurred. Either: • *ObjectQMgrName* specifies the name of a local queue, but it does not have a **Usage** attribute of MQUS_TRANSMISSION. • The *ObjectName* or *ObjectQMgrName* field in the object descriptor specifies the name of a local definition of a remote queue but one of the following applies to the **XmitQName** attribute of the definition: – *XmitQName* is not blank, but specifies a queue that does not have a **Usage** attribute of MQUS_TRANSMISSION – *XmitQName* is blank, but *RemoteQMgrName* specifies a queue that does not have a **Usage** attribute of MQUS_TRANSMISSION • The queue name is resolved through a cell directory, and the remote queue manager name obtained from the cell directory is the name of a local queue, but it does not have a **Usage** attribute of MQUS_TRANSMISSION.
Event type:	Remote.

Platforms:	All.
Event queue:	SYSTEM.ADMIN.QMGR.EVENT.

Event data

QMgrName

Description:	Name of the queue manager generating the event.
Identifier:	MQCA_Q_MGR_NAME.
Data type:	MQCFST.
Maximum length:	MQ_Q_MGR_NAME_LENGTH.
Returned:	Always.

QName

Description:	Queue name from object descriptor (MQOD).
Identifier:	MQCA_Q_NAME.
Data type:	MQCFST.
Maximum length:	MQ_Q_NAME_LENGTH.
Returned:	Always.

XmitQName

Description:	Transmission queue name.
Identifier:	MQCA_XMIT_Q_NAME.
Data type:	MQCFST.
Maximum length:	MQ_Q_NAME_LENGTH.
Returned:	Always.

ApplType

Description:	Type of application making the MQI call that caused the event.
Identifier:	MQIA_APPL_TYPE.
Data type:	MQCFIN.
Returned:	Always.

ApplName

Description:	Name of the application making the MQI call that caused the event.
Identifier:	MQCACF_APPL_NAME.
Data type:	MQCFST.
Maximum length:	MQ_APPL_NAME_LENGTH.
Returned:	Always.

ObjectQMgrName

Description:	Name of the object queue manager.
Identifier:	MQCACF_OBJECT_Q_MGR_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: If the *ObjectName* in the object descriptor (MQOD), when the object was opened, is not the queue manager currently connected.

ConnName

Description: Connection name for client connection.

Identifier: MQCACH_CONNECTION_NAME.

Data type: MQCFST.

Maximum length: MQ_CONN_NAME_LENGTH.

Returned: If the application making the MQI call that caused the event is a client attached application.

ChannelName

Description: Channel name for client connection.

Identifier: MQCACH_CHANNEL_NAME.

Data type: MQCFST.

Maximum length: MQ_CHANNEL_NAME_LENGTH

Returned: If the application making the MQI call that caused the event is a client attached application.

Unknown Alias Base Queue

Event name:	Unknown Alias Base Queue.
Reason code in MQCFH:	"2082 (0822) (RC2082): MQRC_UNKNOWN_ALIAS_BASE_Q" on page 1189. Unknown alias base queue or topic.
Event description:	An MQOPEN or MQPUT1 call was issued specifying an alias queue as the destination, but the *BaseObjectName* in the alias queue attributes is not recognized as a queue or topic name.
Event type:	Local.
Platforms:	All.
Event queue:	SYSTEM.ADMIN.QMGR.EVENT.

Event data

QMgrName

Description: Name of the queue manager generating the event.

Identifier: MQCA_Q_MGR_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: Always.

QName

Description: Queue name from object descriptor (MQOD).

Identifier: MQCA_Q_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_NAME_LENGTH.

Returned: Always.

BaseObjectName

Description: Object name to which the alias resolves.

Identifier: MQCA_BASE_OBJECT_NAME. For compatibility with existing applications, you can still use MQCA_BASE_Q_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_NAME_LENGTH.

Returned: Always.

ApplType

Description: Type of application making the MQI call that caused the event.

Identifier: MQIA_APPL_TYPE.

Data type: MQCFIN.

Returned: Always.

ApplName

Description: Name of the application making the MQI call that caused the event.

Identifier: MQCACF_APPL_NAME.

Data type: MQCFST.

Maximum length: MQ_APPL_NAME_LENGTH.

Returned: Always.

ObjectQMgrName

Description: Name of the object queue manager.

Identifier: MQCACF_OBJECT_Q_MGR_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: If the *ObjectName* in the object descriptor (MQOD), when the object was opened, is not the queue manager currently connected.

BaseType

Description: Type of object to which the alias resolves.

Identifier: MQIA_BASE_TYPE.

Data type: MQCFIN.

Values:

MQOT_Q
Base object type is a queue

MQOT_TOPIC
Base object type is a topic

Returned: Always.

ConnName

Description: Connection name for client connection.

Identifier: MQCACH_CONNECTION_NAME.

Data type: MQCFST.

Maximum length: MQ_CONN_NAME_LENGTH.

Returned: If the application making the MQI call that caused the event is a client attached application.

ChannelName

Description: Channel name for client connection.

Identifier: MQCACH_CHANNEL_NAME.

Data type: MQCFST.

Maximum length: MQ_CHANNEL_NAME_LENGTH

Returned: If the application making the MQI call that caused the event is a client attached application.

Unknown Default Transmission Queue

Event name:	Unknown Default Transmission Queue.
Reason code in MQCFH:	"2197 (0895) (RC2197): MQRC_UNKNOWN_DEF_XMIT_Q" on page 1229. Unknown default transmission queue.
Event description:	An MQOPEN or MQPUT1 call was issued specifying a remote queue as the destination. If a local definition of the remote queue was specified, or if a queue manager alias is being resolved, the **XmitQName** attribute in the local definition is blank. No queue is defined with the same name as the destination queue manager. The queue manager has therefore attempted to use the default transmission queue. However, the name defined by the **DefXmitQName** queue manager attribute is not the name of a locally-defined queue.
Event type:	Remote.
Platforms:	All.
Event queue:	SYSTEM.ADMIN.QMGR.EVENT.

Event data

QMgrName

Description: Name of the queue manager generating the event.

Identifier: MQCA_Q_MGR_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: Always.

QName

Description: Queue name from object descriptor (MQOD).

Identifier: MQCA_Q_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_NAME_LENGTH.

Returned: Always.

XmitQName

Description: Default transmission queue name.

Identifier: MQCA_XMIT_Q_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_NAME_LENGTH.

Returned: Always.

ApplType

Description: Type of application attempting to open the remote queue.

Identifier: MQIA_APPL_TYPE.

Data type: MQCFIN.

Returned: Always.

ApplName

Description: Name of the application attempting to open the remote queue.

Identifier: MQCACF_APPL_NAME.

Data type: MQCFST.

Maximum length: MQ_APPL_NAME_LENGTH.

Returned: Always.

ObjectQMgrName

Description: Name of the object queue manager.

Identifier: MQCACF_OBJECT_Q_MGR_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: If the *ObjectName* in the object descriptor (MQOD), when the object was opened, is not the queue manager currently connected.

ConnName

Description: Connection name for client connection.

Identifier: MQCACH_CONNECTION_NAME.

Data type: MQCFST.

Maximum length: MQ_CONN_NAME_LENGTH.

Returned: If the application making the MQI call that caused the event is a client attached application.

ChannelName

Description: Channel name for client connection.

Identifier: MQCACH_CHANNEL_NAME.

Data type: MQCFST.

Maximum length: MQ_CHANNEL_NAME_LENGTH

Returned: If the application making the MQI call that caused the event is a client attached application.

Unknown Object Name

Event name:	Unknown Object Name.
Reason code in MQCFH:	"2085 (0825) (RC2085): MQRC_UNKNOWN_OBJECT_NAME" on page 1190. Unknown object name.
Event description:	On an MQOPEN or MQPUT1 call, the *ObjectQMgrName* field in the object descriptor MQOD is set to one of the following options. It is either: • Blank • The name of the local queue manager • The name of a local definition of a remote queue (a queue manager alias) in which the **RemoteQMgrName** attribute is the name of the local queue manager However, the *ObjectName* in the object descriptor is not recognized for the specified object type.
Event type:	Local.
Platforms:	All.
Event queue:	SYSTEM.ADMIN.QMGR.EVENT.

Event data

QMgrName

Description: Name of the queue manager generating the event.

Identifier: MQCA_Q_MGR_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: Always.

ApplType

Description: Type of application making the MQI call that caused the event.

Identifier: MQIA_APPL_TYPE.

Data type: MQCFIN.

Returned: Always.

ApplName

Description: Name of the application making the MQI call that caused the event.

Identifier: MQCACF_APPL_NAME.

Data type: MQCFST.

Maximum length: MQ_APPL_NAME_LENGTH.

Returned: Always.

QName

Description: Queue name from object descriptor (MQOD).

Identifier: MQCA_Q_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_NAME_LENGTH.

Returned: If the object opened is a queue object. Either *QName* or *TopicName* is returned.

ProcessName

Description: Process object name from object descriptor (MQOD).

Identifier: MQCA_PROCESS_NAME.

Data type: MQCFST.

Maximum length: MQ_PROCESS_NAME_LENGTH.

Returned: If the object opened is a process object. One of *ProcessName*, *QName*, or *TopicName* is returned.

ObjectQMgrName

Description: Name of the object queue manager.

Identifier: MQCACF_OBJECT_Q_MGR_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: If the *ObjectName* in the object descriptor (MQOD), when the object was opened, is not the queue manager currently connected.

TopicName

Description: Topic object name from object descriptor (MQOD).

Identifier: MQCA_TOPIC_NAME.

Data type: MQCFST.

Maximum length: MQ_TOPIC_NAME_LENGTH.

Returned: If the object opened is a topic object. One of *ProcessName*, *QName*, or *TopicName* is returned.

ConnName

Description:	Connection name for client connection.
Identifier:	MQCACH_CONNECTION_NAME.
Data type:	MQCFST.
Maximum length:	MQ_CONN_NAME_LENGTH.
Returned:	If the application making the MQI call that caused the event is a client attached application.

ChannelName

Description:	Channel name for client connection.
Identifier:	MQCACH_CHANNEL_NAME.
Data type:	MQCFST.
Maximum length:	MQ_CHANNEL_NAME_LENGTH
Returned:	If the application making the MQI call that caused the event is a client attached application.

Unknown Remote Queue Manager

Event name:	Unknown Remote Queue Manager.
Reason code in MQCFH:	“2087 (0827) (RC2087): MQRC_UNKNOWN_REMOTE_Q_MGR” on page 1191. Unknown remote queue manager.

Event description:	On an MQOPEN or MQPUT1 call, an error occurred with queue-name resolution, for one of the following reasons: • *ObjectQMgrName* is either blank or the name of the local queue manager, and *ObjectName* is the name of a local definition of a remote queue that has a blank *XmitQName*. However, there is no (transmission) queue defined with the name of *RemoteQMgrName*, and the **DefXmitQName** queue manager attribute is blank. • *ObjectQMgrName* is the name of a queue manager alias definition (held as the local definition of a remote queue) that has a blank *XmitQName*. However, there is no (transmission) queue defined with the name of *RemoteQMgrName*, and the **DefXmitQName** queue manager attribute is blank. • *ObjectQMgrName* specified is not: – Blank – The name of the local queue manager – The name of a local queue – The name of a queue manager alias definition (that is, a local definition of a remote queue with a blank *RemoteQName*) and the **DefXmitQName** queue manager attribute is blank. • *ObjectQMgrName* is blank or is the name of the local queue manager, and *ObjectName* is the name of a local definition of a remote queue (or an alias to one), for which *RemoteQMgrName* is either blank or is the name of the local queue manager. This error occurs even if the *XmitQName* is not blank. • *ObjectQMgrName* is the name of a local definition of a remote queue. In this case, it should be a queue manager alias definition, but the *RemoteQName* in the definition is not blank. • *ObjectQMgrName* is the name of a model queue. • The queue name is resolved through a cell directory. However, there is no queue defined with the same name as the remote queue manager name obtained from the cell directory. Also, the **DefXmitQName** queue manager attribute is blank. • On z/OS only: a message was put to a queue manager in a queue sharing group and *SQQMNAME* is set to USE. This routes the message to the specified queue manager in order to be put on the queue. If *SQQMNAME* is set to IGNORE, the message is put to the queue directly.
Event type:	Remote.
Platforms:	All.
Event queue:	SYSTEM.ADMIN.QMGR.EVENT.

Event data

QMgrName

Description:	Name of the queue manager generating the event.
Identifier:	MQCA_Q_MGR_NAME.
Data type:	MQCFST.
Maximum length:	MQ_Q_MGR_NAME_LENGTH.
Returned:	Always.

QName

Description: Queue name from object descriptor (MQOD).

Identifier: MQCA_Q_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_NAME_LENGTH.

Returned: Always.

ApplType

Description: Type of application attempting to open the remote queue.

Identifier: MQIA_APPL_TYPE.

Data type: MQCFIN.

Returned: Always.

ApplName

Description: Name of the application attempting to open the remote queue.

Identifier: MQCACF_APPL_NAME.

Data type: MQCFST.

Maximum length: MQ_APPL_NAME_LENGTH.

Returned: Always.

ObjectQMgrName

Description: Name of the object queue manager.

Identifier: MQCACF_OBJECT_Q_MGR_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: If the *ObjectName* in the object descriptor (MQOD), when the object was opened, is not the queue manager currently connected.

ConnName

Description: Connection name for client connection.

Identifier: MQCACH_CONNECTION_NAME.

Data type: MQCFST.

Maximum length: MQ_CONN_NAME_LENGTH.

Returned: If the application making the MQI call that caused the event is a client attached application.

ChannelName

Description: Channel name for client connection.

Identifier: MQCACH_CHANNEL_NAME.

Data type: MQCFST.

Maximum length: MQ_CHANNEL_NAME_LENGTH

Returned:	If the application making the MQI call that caused the event is a client attached application.

Unknown Transmission Queue

Event name:	Unknown Transmission Queue.
Reason code in MQCFH:	"2196 (0894) (RC2196): MQRC_UNKNOWN_XMIT_Q" on page 1229. Unknown transmission queue.
Event description:	On an MQOPEN or MQPUT1 call, a message is to be sent to a remote queue manager. The *ObjectName* or the *ObjectQMgrName* in the object descriptor specifies the name of a local definition of a remote queue (in the latter case queue manager aliasing is being used). However, the **XmitQName** attribute of the definition is not blank and not the name of a locally-defined queue.
Event type:	Remote.
Platforms:	All.
Event queue:	SYSTEM.ADMIN.QMGR.EVENT.

Event data

QMgrName

Description:	Name of the queue manager generating the event.
Identifier:	MQCA_Q_MGR_NAME.
Data type:	MQCFST.
Maximum length:	MQ_Q_MGR_NAME_LENGTH.
Returned:	Always.

QName

Description:	Queue name from object descriptor (MQOD).
Identifier:	MQCA_Q_NAME.
Data type:	MQCFST.
Maximum length:	MQ_Q_NAME_LENGTH.
Returned:	Always.

XmitQName

Description:	Transmission queue name.
Identifier:	MQCA_XMIT_Q_NAME.
Data type:	MQCFST.
Maximum length:	MQ_Q_NAME_LENGTH.
Returned:	Always.

ApplType

Description:	Type of application making the MQI call that caused the event.
Identifier:	MQIA_APPL_TYPE.

Data type: MQCFIN.

Returned: Always.

ApplName

Description: Name of the application making the MQI call that caused the event.

Identifier: MQCACF_APPL_NAME.

Data type: MQCFST.

Maximum length: MQ_APPL_NAME_LENGTH.

Returned: Always.

ObjectQMgrName

Description: Name of the object queue manager.

Identifier: MQCACF_OBJECT_Q_MGR_NAME.

Data type: MQCFST.

Maximum length: MQ_Q_MGR_NAME_LENGTH.

Returned: If the *ObjectName* in the object descriptor (MQOD), when the object was opened, is not the queue manager currently connected.

ConnName

Description: Connection name for client connection.

Identifier: MQCACH_CONNECTION_NAME.

Data type: MQCFST.

Maximum length: MQ_CONN_NAME_LENGTH.

Returned: If the application making the MQI call that caused the event is a client attached application.

ChannelName

Description: Channel name for client connection.

Identifier: MQCACH_CHANNEL_NAME.

Data type: MQCFST.

Maximum length: MQ_CHANNEL_NAME_LENGTH

Returned: If the application making the MQI call that caused the event is a client attached application.

Messages and reason codes

You can use the following messages and reason codes to help you solve problems with your IBM MQ components or applications.

Multi AMQ messages on Multiplatforms

The AMQ series of diagnostic messages are listed in this section in numerical order, grouped according to the part of IBM MQ from which they originate.

Note: If a message is specific to a single platform, this is indicated after the message identifier. Although some messages are listed several times, each instance relates to a different platform. If present, the

version common to a number of platforms is listed first, followed by versions for individual platforms. Ensure that you read the appropriate version.

For details of these messages, see IBM Documentation:

Reading a message

The following information is provided for each message:

Message identifier
The message identifier is in three parts:

1. The characters "AMQ", which identify the message as being from IBM MQ.
2. A four-digit decimal code.
3. A suffix letter, indicating the severity of the message (I, W, E, S or T), as shown in the Severity list.

   The suffix letter is included by default. You can disable the suffix letter, for example, if you have scripts that are dependent on the old message format that did not include the suffix letter, by setting the AMQ_DIAGNOSTIC_MSG_SEVERITY environment variable to 0.

Message text
A summary of the message

Severity

0: Informational
10: Warning
20: Error
30: Error
40: Severe
50: Terminating

Explanation
An explanation of the message giving further information.

Response
The response required from the user. In some cases, particularly for information messages, this might be "No action is required".

Message variables

Some messages display text or numbers that vary according to the circumstances giving rise to the message; these are known as *message variables*. The message variables are indicated as <insert_1>, <insert_2>, and so on.

In some cases a message might have variables in the Explanation or Response. Find the values of the message variables by looking in the error log. The complete message, including the Explanation and the Response, is recorded there.

Related concepts
“IBM MQ for z/OS messages, completion, and reason codes” on page 289
Use this topic to interpret and understand the messages and codes issued by IBM MQ for z/OS.

Related tasks
Using error logs

Related reference
“API completion and reason codes” on page 1151
For each call, a completion code and a reason code are returned by the queue manager or by an exit routine, to indicate the success or failure of the call.

“PCF reason codes” on page 1360

Reason codes might be returned by a broker in response to a command message in PCF format, depending on the parameters used in that message.

"Transport Layer Security (TLS) return codes" on page 1441
IBM MQ can use TLS with the various communication protocols. Use this topic to identify the error codes that can be returned by TLS.

"WCF custom channel exceptions" on page 1446
Diagnostic messages are listed in this topic in numeric order, grouped according to the part of the Microsoft Windows Communication Foundation (WCF) custom channel from which they originate.

IBM MQ Console messages

Reference information to help you identify and interpret the messages for the IBM MQ Console.

For details of these messages, see IBM Documentation:

Messages for Long Term Support (LTS) releases	Messages for Continuous Delivery (CD) releases
• MQWB20xx: IBM MQ Console messages	• MQWB20xx: IBM MQ Console messages

Reading a message

The following information is provided for each message:

Message identifier
The message identifier is in three parts:

1. The characters "MQWB", which identify the message as being from the IBM MQ Console
2. A four-digit decimal code
3. A suffix letter, indicating the severity of the message (I, W, E, S or T). See the following severity list.

Message text
A summary of the message

Severity

0: Informational
10: Warning
20: Error
30: Error
40: Severe
50: Terminating

Explanation
An explanation of the message giving further information.

Response
The response required from the user. In some cases, particularly for information messages, this might be "No action is required".

Message variables

Some messages display text or numbers that vary according to the circumstances giving rise to the message; these are known as *message variables*. The message variables are indicated as <insert_1>, <insert_2>, and so on.

In some cases a message might have variables in the Explanation or Response. Find the values of the message variables by looking in the error log. The complete message, including the Explanation and the Response, is recorded there.

IBM MQ Internet Pass-Thru messages

Reference information to help you identify and interpret the messages for IBM MQ Internet Pass-Thru.

When run from the command line, IBM MQ Internet Pass-Thru (MQIPT) displays information, warning, and error messages on the console.

MQIPT diagnostic messages are listed in this section in numerical order, grouped according to the part of MQIPT from which they originate.

Messages for Long Term Support (LTS) releases	Messages for Continuous Delivery (CD) releases
• MQCAxxxx: administration client messages • MQCPxxxx: MQIPT messages	• MQCAxxxx: administration client messages • MQCPxxxx: MQIPT messages

Reading a message

All message identifiers follow the same format:

```
MQCpsnnn
```

where:

- *p* is the producer of the message:
  - A: The **mqiptAdmin** command
  - P: MQIPT
- *s* is the severity of the message:
  - I: information
  - W: warning
  - E: error
- *nnn* is the three-digit message number.

Message variables

Some messages display text or numbers that vary according to the circumstances giving rise to the message; these are known as *message variables*. The message variables are indicated as <insert_1>, <insert_2>, and so on.

MQCAxxxx: administration client messages (LTS)

MQCAE001 Unknown host: <insert_1>

Explanation:
The MQIPT host cannot be found.

User response:
Check the host name was specified correctly. Try to PING the host name or use its IP address.

MQCAE002 The following error was reported by the system: <insert_1>

Explanation:
An error has occurred while communicating with MQIPT.

User response:
Review the text of the error message and take the appropriate action.

MQCAE023 MQIPT did not recognize your password

Explanation:
An incorrect password was specified when issuing an administrative command to MQIPT.

User response:
Ensure that the correct password is entered when issuing administrative commands to MQIPT.

MQCAE024 MQIPT <insert_1> has not recognized the command

Explanation:
An error has been detected while communicating with the MQIPT.

User response:

Ensure that the **mqiptAdmin** command is at the same version or higher than the MQIPT instance being administered. If the condition persists, contact IBM Software Support.

MQCAE026 **Remote shutdown is disabled on MQIPT <insert_1>**

Explanation:
MQIPT has received a request to shut down but cannot proceed because remote shutdown is not enabled.

User response:
To enable remote shutdown of MQIPT, edit the configuration file and set the **RemoteShutDown** property to `true`.

MQCAE057 **Invalid port number specified**

Explanation:
An invalid port number was specified as a parameter to a command.

User response:
Specify a valid port number in the command parameters.

MQCAE058 **No local instance of MQIPT found with the name <insert_1>**

Explanation:
No active local instance of MQIPT with the specified name was found by the **mqiptAdmin** command.

User response:
Specify the name of an active instance of MQIPT as a parameter to the **mqiptAdmin** command. The instance of MQIPT must be running on the local machine under the same user ID as the **mqiptAdmin**. Alternatively, on AIX and Linux, **mqiptAdmin** can be run as `root`. To list active local instances of MQIPT, use the **-list** option of the **mqiptAdmin** command.

MQCAE059 **MQIPT <insert_1> is already shutting down**

Explanation:
A command was sent to MQIPT, but the MQIPT instance is already shutting down.

MQCAE060 **Local administration is disabled**

Explanation:
A command was sent to local instance of MQIPT, but local administration is disabled.

User response:
Either enable local administration using the **LocalAdmin** property, or administer MQIPT using the command port.

MQCAE061 **Error establishing TLS connection to MQIPT: <insert_1>**

Explanation:
An error occurred when the **mqiptAdmin** command tried to establish a TLS connection to MQIPT. Details of the exception that was thrown follow this message.

User response:
Examine the details of the exception in the message. Ensure that the TLS configuration for the **mqiptAdmin** command is correct.

- The properties file must contain the name of the trust store to be used by **mqiptAdmin** to validate the identity of the remote MQIPT instance, and the password to access the trust store.
- The trust store used by **mqiptAdmin** must contain the the CA certificate of the CA that signed the MQIPT server certificate.

MQCAE062 **The file <insert_1> specified in the property <insert_2> does not exist or cannot be read.**

Explanation:
A property in the **mqiptAdmin** properties file refers to a file or directory which does not exist.

User response:
Edit the properties file and specify the correct location for the file or directory.

MQCAE063 **Properties file <insert_1> does not exist or cannot be read.**

Explanation:
The properties file specified with the **-p** parameter to **mqiptAdmin** does not exist or cannot be read.

User response:
Ensure that the name of the properties file is specified correctly with the **-p** parameter, and that the file can be read by the user that is running the **mqiptAdmin** command.

MQCAE064 **Error reading properties file <insert_1>.**

Explanation:
An error occurred when reading the properties file specified with the **-p** parameter to **mqiptAdmin**.

User response:
Ensure that the name of the properties file is specified correctly with the **-p** parameter, and that the file is can be read by the user that is running the **mqiptAdmin** command.

MQCAE065 **MQIPT <insert_1> has shut down.**

Explanation:
MQIPT has shut down while processing a command issued by **mqiptAdmin**.

MQCAE066 **MQIPT command level <insert_1> is not supported by mqiptAdmin version <insert_2>.**

Explanation:
The **mqiptAdmin** command is connected to an instance of MQIPT which is at a command level that is not supported by this version of **mqiptAdmin**.

User response:
Ensure that the **mqiptAdmin** command is always at the same version, or a higher version, than the instance of MQIPT it is connected to.

MQCAI025 **MQIPT <insert_1> has been refreshed**

Explanation:
MQIPT has been updated by reading its configuration file.

MQCAI026 **MQIPT <insert_1> has received shutdown request**

Explanation:
MQIPT has acknowledged receipt of a shutdown request and will now shut down.

MQCAI105 **Sending <insert_1> command to MQIPT instance with name <insert_2>**

Explanation:
mqiptAdmin is sending the specified command to the local instance of MQIPT with the specified name.

MQCAI106 **Local active instances of MQIPT:**

Explanation:
This message is displayed by the **mqiptAdmin** command to list the local active instances of MQIPT.

MQCAI107 **No local active instances of MQIPT**

Explanation:
The **mqiptAdmin** command found no local active instances of MQIPT running under the same user ID as the **mqiptAdmin** command.

MQCAI108 **If the command port is secured with TLS, you must specify the -s parameter to enable a secure connection.**

Explanation:
The **mqiptAdmin** command connected to MQIPT using an unsecured connection, but received a response that indicated that the command port is secured with TLS.

User response:
To connect to MQIPT using TLS, specify the **-s** parameter to the **mqiptAdmin** command.

MQCAI109 **The connection to MQIPT is secured with <insert_1>.**

Explanation:
The **mqiptAdmin** command has connected to MQIPT. The connection is secured using the indicated protocol.

MQCAW001 **The connection to MQIPT is unsecured.**

Explanation:
The **mqiptAdmin** command has connected to MQIPT using an unsecured connection.

User response:
None required. To connect to MQIPT securely using TLS, specify the **-s** parameter to the **mqiptAdmin** command.

MQCPxxxx: MQIPT messages (LTS)

MQCPA104 **Command complete from MQIPT server at <insert_1>**

Explanation:
A command sent by the **mqiptAdmin** command has been accepted and run by MQIPT.

MQCPE001 **Directory does not exist or is not a directory <insert_1>**

Explanation:
At MQIPT initialization, a required directory could not be found. This message refers to a directory specified either in the MQIPT configuration file `mqipt.conf` or in the MQIPT command line startup options on the default directory.

User response:
Specify the correct directory and retry the command.

MQCPE004 **Route startup failed on port <insert_1>**

Explanation:
It was not possible to start the route with the specified **ListenerPort** number.

User response:
An I/O error occurred during route startup. Check for other adjacent error messages and log records to provide further explanation of the problem.

MQCPE005 **The configuration file <insert_1> could not be found**

Explanation:
The MQIPT configuration file `mqipt.conf` could not be found in the specified directory

User response:
Specify the correct directory and retry the command.

MQCPE006 **The number of routes has exceeded <insert_1>. MQIPT will start but this configuration is unsupported**

Explanation:
Your configuration has exceeded the maximum supported number of routes for one instance of MQIPT. Operation will not be halted but the system might become unstable or overloaded as a result. Configurations that exceed the stated maximum number of routes will not be supported.

User response:
Consider starting additional instances of MQIPT with fewer routes per instance.

MQCPE007 **Route not restarted on listener port <insert_1>**

Explanation:
During a refresh operation, the route that was operating on the specified **ListenerPort** was not restarted with the new configuration.

User response:
Check for other adjacent error messages for further explanation of the problem.

MQCPE008 **Duplicate route defined for listener port <insert_1>**

Explanation:
More than one route has been defined with the same **ListenerPort** value.

User response:
Remove the duplicate route from the configuration file and retry the command.

MQCPE009 **Log directory <insert_1> is not valid**

Explanation:
The indicated log directory either does not exist or is not accessible.

User response:
Check the directory exists and is accessible by MQIPT.

MQCPE010 **Listener or command port number <insert_1> is not valid**

Explanation:
The port address supplied for the command port or listener port parameter is invalid.

User response:
Specify a valid port address that is free for use. For guidance on use of port addresses in your network, consult your network administrator.

MQCPE012 **The value <insert_1> is not valid for the property <insert_2>**

Explanation:
An invalid property value has been specified.

User response:
Refer to IBM MQ Internet Pass-Thru configuration reference for full details of the valid values for each property.

MQCPE013 **ListenerPort property was not found in route <insert_1>**

Explanation:
MQIPT has detected a route in the configuration file that does not contain a **ListenerPort** property. The **ListenerPort** property is the primary and unique identifier for each route, and is therefore mandatory.

User response:
Specify a valid **ListenerPort** property for the given route.

MQCPE014 **ListenerPort property value <insert_1> is not valid**

Explanation:
An invalid port address is specified for the **ListenerPort** property of a route.

User response:
Specify a port number in the range 0 to 65535 for each **ListenerPort** property in the MQIPT configuration file.

MQCPE015 **No text was found for message number <insert_1>**

Explanation:
An internal error has been encountered for which no description is available.

User response:
The `mqipt.properties` file may have become corrupted and the specified message number could not be found. Check that the `mqipt.properties` file is in the `com.ibm.mq.ipt.jar` file, and that the message number is in the `mqipt.properties` file. If you are using the `MQIPT_PATH` environment variable, ensure it is set correctly.

MQCPE016 **The maximum number of connection threads is <insert_1> but this is less than the minimum number of connection threads, which is <insert_2>**

Explanation:
Your configuration file has specified the minimum number of connection threads with a value greater than the maximum number of connection threads.

User response:
This could be an error in a single route, a conflict between a global property and a route property, or a route property overriding the system default values.

Refer to IBM MQ Internet Pass-Thru configuration reference for full details of the valid values and applicable defaults for each property.

MQCPE017 **The exception <insert_1> was thrown causing MQIPT to shut down**

Explanation:
MQIPT has abnormally terminated and has been shut down. This may have occurred because of system environmental conditions or constraints, such as memory overflow.

User response:
If the condition persists, contact IBM Software Support.

MQCPE018 **The ListenerPort property is blank - the route will not start**

Explanation:
The **ListenerPort** number has been omitted in a route.

User response:
Edit the configuration file and add a valid **ListenerPort**.

MQCPE019 **The stanza <insert_1> was not found before the following : <insert_2>**

Explanation:
A sequence error has occurred in the configuration file.

User response:
Edit the configuration file and make sure that all `[route]` stanzas are after the `[global]` stanza.

MQCPE020 **The new value for MaxConnectionThreads is <insert_1>. This must be greater than the current value <insert_2>**

Explanation:
After the route has started, the **MaxConnectionThread** property can only be increased.

User response:
Edit the configuration file and change the **MaxConnectionThread** property.

MQCPE021 **The Destination property was not supplied for route <insert_1>**

Explanation:
The **Destination** property is mandatory for a route, but was omitted in the route specified.

User response:
Edit the configuration file and add a **Destination** property for the given route.

MQCPE022 **The <insert_1> value <insert_2> is outside the valid range 1 - 65535**

Explanation:
The value of the **CommandPort** or **SSLCommandPort** property is outside the range 1-65535.

User response:
Edit the configuration file and change the value of the indicated property to a valid port number.

MQCPE023 **Request to shut down MQIPT from the Administration Client at <insert_1> is ignored because remote shutdown is disabled**

Explanation:
An attempt to shut down the MQIPT remotely has failed because remote shutdown was not enabled in the configuration file.

User response:
To enable remote shutdown of MQIPT, edit the configuration file and set the **RemoteShutDown** property to `true`.

MQCPE024 **The command received by the MQIPT controller has not been recognized**

Explanation:
The MQIPT command port received a command that it does not recognize.

User response:
Check the `mqipt.log` file for the identity of the command.

MQCPE025 **Failed to connect to server on host <insert_1>, port <insert_2>: <insert_3>**

Explanation:
The **mqiptAdmin** command could not connect to MQIPT at the indicated network address and port number.

User response:
Ensure that the MQIPT command port is configured to listen on the indicated port, and that MQIPT is running on indicated host. Check the MQIPT console log for any errors that occurred when starting the command server.

MQCPE026 **No reply received from server on host <insert_1>, port <insert_2>**

Explanation:
The **mqiptAdmin** command did not receive a reply from MQIPT. This can occur if the request times-out, or if there is a problem with MQIPT.

User response:

Check the MQIPT console log and connection log for additional error messages.

MQCPE027 Reply from MQIPT not recognized

Explanation:
The **mqiptAdmin** command received a reply from MQIPT which it did not recognize.

User response:
Ensure that the **mqiptAdmin** command is at the same version, or a higher version, than the instance of MQIPT it is connecting to.

MQCPE028 Invalid stanza detected : <insert_1>

Explanation:
The stated unrecognized stanza has been found in the configuration file.

User response:
Only `[global]` and `[route]` stanzas are valid in the configuration file.

MQCPE029 Was not able to flush log output

Explanation:
Some messages might not have been written to the log because the communication buffer could not be flushed.

User response:
Check that the MQIPT home directory file system is not full and that MQIPT still has access to write to the `logs` subdirectory.

MQCPE034 Administration Client at <insert_1> did not supply the correct password

Explanation:
The **AccessPW** property in the configuration file did not match that provided by the Administration Client.

User response:
Either change the **AccessPW** property in the configuration file or the saved password in the Administration Client.

MQCPE035 Failed to start command listener on port <insert_1>

Explanation:
An I/O error occurred starting the command port listener on the specified port.

User response:
Check the port number used for the **CommandPort** property in the configuration file.

MQCPE042 There is a conflict with the following properties on route <insert_1> :

Explanation:
Some properties can not be used with others. This message precedes the list of properties in conflict.

User response:
Check the following error messages and take the appropriate action.

MQCPE043 ....<insert_1> and <insert_2>

Explanation:
The two specified properties cannot both be set at the same time on the same route.

User response:
Edit the configuration file and remove one of the specified properties on the given route.

MQCPE045 ....HTTP proxy or server name is missing

Explanation:
The **HTTPProxy** or **HTTPServer** property must be set if the **HTTP** property has been set to `true`.

User response:
Edit the configuration file and define an **HTTPProxy** or **HTTPServer** for the given route.

MQCPE048 Route startup failed on port <insert_1>, exception was : <insert_2>

Explanation:
It was not possible to start the route with the specified **ListenerPort** number.

User response:
Check for other adjacent error messages and log records to provide further explanation of the problem.

MQCPE055 ....Socks proxy name is missing

Explanation:
The **SocksProxy** property must be set if the **SocksClient** property has been set to true.

User response:
Edit the configuration file and define a **SocksProxy** for the given route.

MQCPE056 Conflict with route properties

Explanation:
Some properties cannot be used with others.

User response:
Check the console messages for details of the error and take the appropriate action.

MQCPE057 SSL protocol (<insert_1>) was not recognized

Explanation:
The route has been put into SSL/TLS proxy mode and the initial data flow is not recognized.

User response:

Make sure only SSL/TLS connections are being made to this route.

MQCPE058 **CONNECT request to <insert_3>(<insert_4>) through <insert_1>(<insert_2>) failed**

Explanation:
An HTTP CONNECT request was sent to the HTTP proxy to create an SSL tunnel to the HTTP server. The HTTP proxy did not send back a "200 OK" response to this request.

User response:
This can be caused by various problems. Enable tracing on the route and retry the connection. The trace file will show the real error.

MQCPE059 **There are no defined key ring files**

Explanation:
An SSL client or server has been defined without specifying at least one key ring file.

User response:
Use the **SSLClientKeyRing** and **SSLClientCAKeyRing** properties on the client side, or **SSLServerKeyRing** and **SSLServerCAKeyRing** on the server side, to define a key ring file and then restart the route.

MQCPE060 **Runtime error setting SSL client connect timeout to <insert_1> seconds**

Explanation:
An SSL runtime error has occurred on the client side setting the timeout value.

User response:
Check the value specified in the **SSLClientConnectTimeout** property is valid. Running a trace on the given route will show further error information.

MQCPE061 **There are no enabled cipher suites**

Explanation:
An SSL client or server connection has been started but MQIPT is unable to determine a valid cipher suite.

User response:
Check there are valid certificates in the defined key ring file(s). The private and public keys used to generate the certificates and the encryption algorithms used must comply with the list of supported cipher suites. See CipherSuites supported by MQIPT for the list of cipher suites supported by MQIPT.

MQCPE062 **Runtime error setting SSL cipher suite <insert_1>**

Explanation:
An unsupported SSL cipher suite has been defined on the client or server side.

User response:
Check the value specified in the **SSLClientCipherSuites** or **SSLServerCipherSuites** is valid and supported on this connection. Running a trace on the given route will show the list of enabled cipher suites. See CipherSuites supported by MQIPT for the list of cipher suites supported by MQIPT.

MQCPE063 **File <insert_1> already exists - use the replace option**

Explanation:
The file name parameter specified for the **mqiptPW** command already exists.

User response:
Either choose another file name or use the replace option.

MQCPE064 **Runtime error generating decryption keys :\n <insert_1>**

Explanation:
An error has occurred while generating cipher keys to decrypt the password used to open a key ring file.

User response:
The runtime error listed in the message should be rectified and the command run again.

MQCPE065 **....LDAP server name is missing**

Explanation:
The **LDAPServer1** or **LDAPServer2** property must be set if the **LDAP** property has been set to `true`.

User response:
Edit the configuration file and define an **LDAPServer*** for the given route.

MQCPE066 **....LDAP password is missing for LDAP server <insert_1>**

Explanation:
An LDAP userid has been specified without a password, for either the main or backup LDAP server.

User response:
Edit the configuration file and define an LDAP password for the given route. The **LDAPServer1Password** property is for the main server and **LDAPServer2Password** property is for the backup server.

MQCPE067 **....SSLClient or SSLServer missing for LDAP server**

Explanation:
The **SSLClient** or **SSLServer** property must be set if the LDAP property has been set to true.

User response:

Edit the configuration file and define an **SSLClient** or **SSLServer** for the given route.

MQCPE068 ....Security exit name is missing

Explanation:
The **SecurityExitName** property must be set if the **SecurityExit** property has been set to `true`.

User response:
Edit the configuration file and define a **SecurityExitName** for the given route.

MQCPE071 Error writing to <insert_1>

Explanation:
An error occurred while creating or updating the file containing the encrypted password. The error message also contains the exception thrown.

User response:
This error is generated from the **mqiptPW** command. The error listed in the exception should be rectified and the command run again.

MQCPE072 An unknown error occurred in security exit <insert_1>

Explanation:
An error occurred in a user-defined security exit while validating a connection request.

User response:
Enable tracing in the security exit and try the connection request again. The error will be recorded in the security exit trace file.

MQCPE073 Security exit <insert_1> timed out

Explanation:
A user-defined security exit timed out while validating a connection request.

User response:
Increase the timeout period for the security exit and try the connection request again.

MQCPE074 ....Certificate exit name is missing

Explanation:
The **SSLExitName** property must be set if the **SSLClientExit** or **SSLServerExit** property has been set to `true`.

User response:
Edit the configuration file and define a **SSLExitName** for the given route.

MQCPE075 ....SSLPlainConnections needs SSLServer or SSLProxyMode enabled

Explanation:
The **SSLExitName** property must be set if the **SSLClientExit** or **SSLServerExit** property has been set to `true`.

User response:
Edit the configuration file and define a **SSLExitName** for the given route.

MQCPE076 Route <insert_1> property <insert_2> contains unsupported CipherSuites. The following CipherSuites are unsupported: <insert_3>

Explanation:
At least one unsupported cipher suite was included in the **SSLClientCipherSuites** or **SSLServerCipherSuites** property.

User response:
Edit the configuration file and remove the unsupported cipher suite from the route configuration.

MQCPE077 Route <insert_1> property <insert_2> specifies file location <insert_3> which does not exist.

Explanation:
A route property refers to a file or directory which does not exist.

User response:
Edit the configuration file and specify the correct location for the file or directory.

MQCPE078 Route <insert_1> property <insert_2> specifies file location <insert_3> which cannot be read.

Explanation:
A route property refers to a file that cannot be read.

User response:
Ensure that the file permissions allow MQIPT to read it.

MQCPE079 Route <insert_1> site certificate label <insert_2> was not found in key ring file <insert_3>.

Explanation:
A site certificate label was specified in the definition of the specified route, but a personal certificate with that label was not found in the key ring file.

User response:
Ensure that correct site certificate label is specified, and that a personal certificate with the label exists in the appropriate key ring.

MQCPE080 Unable to determine MQIPT installation directory. Set the MQIPT_PATH environment variable to the absolute path of the top-level MQIPT directory.

Explanation:
The MQIPT command was unable to determine the installation directory.

User response:
Set the MQIPT_PATH environment variable to the absolute path of the top-level MQIPT directory.

MQCPE081 **Invalid MQIPT_PATH <insert_1>. The directory does not exist or does not contain a valid MQIPT installation.**

Explanation:
The MQIPT_PATH environment variable is set incorrectly. Either the directory does not exist or the directory is not an MQIPT installation.

User response:
Check the MQIPT_PATH environment variable is set correctly and re-run the command.

MQCPE082 **Unable to install the MQIPT service because a service is already installed. Only one MQIPT service may be installed at a time.**

Explanation:
The user attempted to install the MQIPT service, but an MQIPT service is already installed. Only one MQIPT service may be installed on the system at a time.

User response:
Merge the required routes into the existing MQIPT service configuration, or remove the existing service and install the new service in its place.

MQCPE083 **Unable to remove the MQIPT service because the installed service was not installed by the current MQIPT installation. Run mqiptService from the MQIPT installation that installed the service.**

Explanation:
The MQIPT service can only be removed using the MQIPT installation that originally installed it. This error occurs when you have multiple MQIPT installations on the system and you attempt to remove the MQIPT service using a different installation from the one that originally installed it.

User response:
Run the command **mqiptService -remove** from the correct MQIPT installation.

MQCPE084 **The MQIPT service is not installed.**

Explanation:
The user attempted to remove the MQIPT service but there is no MQIPT service installed.

MQCPE086 **Security exit <insert_1> for route <insert_2> failed to initialize due to error <insert_3>.**

Explanation:
The security exit initialization method returned an unexpected error, which prevented the route from starting.

User response:
Investigate the cause of the error and restart the route.

MQCPE087 **Security exit <insert_1> for route <insert_2> failed to load due to error <insert_3>.**

Explanation:
The security exit could not be loaded, which prevented the route from starting.

User response:
Investigate the cause of the exit load error and restart the route.

MQCPE088 **Certificate exit <insert_1> for route <insert_2> failed to initialize due to error <insert_3>.**

Explanation:
The certificate exit initialization method returned an unexpected error, which prevented the route from starting.

User response:
Investigate the cause of the error and restart the route.

MQCPE089 **Certificate exit <insert_1> for route <insert_2> failed to load due to error <insert_3>.**

Explanation:
The certificate exit could not be loaded, which prevented the route from starting.

User response:
Investigate the cause of the exit load error and restart the route.

MQCPE090 **The security exit rejected the connection with return code <insert_1> and error <insert_2>.**

Explanation:
The security exit rejected a connection to the route listener port.

User response:
Investigate the error returned by the exit.

MQCPE091 **The SSLClient certificate exit rejected the connection with return code <insert_1> and error <insert_2>.**

Explanation:
The SSL client certificate exit rejected the remote server certificate.

User response:
Investigate the error returned by the exit.

MQCPE092 **The SSLServer certificate exit rejected the connection with return code <insert_1> and error <insert_2>.**

Explanation:
The SSL server certificate exit rejected the remote client certificate.

User response:
Investigate the error returned by the exit.

MQCPE093 **Global property <insert_1> specifies file location <insert_2> which does not exist.**

Explanation:
A global property refers to a file or directory which does not exist.

User response:
Edit the configuration file and specify the correct location for the file or directory.

MQCPE094 **Global property <insert_1> specifies file location <insert_2> which cannot be read.**

Explanation:
A global property refers to a file that cannot be read.

User response:
Ensure that the file permissions allow MQIPT to read it.

MQCPE095 **The MQIPT installation directory <insert_1> must not contain a space on this platform.**

Explanation:
The MQIPT installation directory contains a space character, which is not supported on AIX or Linux.

User response:
Rename the installation directory so that it does not contain a space.

MQCPE096 **Error enabling TCP keep alive**

Explanation:
The TCP keep alive route property is set, but MQIPT was unable to enable TCP keep alive.

User response:
Investigate the cause of the failure or disable TCP keep alive.

MQCPE097 **....SSLClient needs to be true for HTTPS communication**

Explanation:
The **SSLClient** property must be set to `true` if **HTTPS** has been set to `true`.

User response:
Edit the configuration file and define **SSLClient** as `true` for the given route.

MQCPE098 **....HTTPS needs to be true when SSLClient and HTTP are both set to true**

Explanation:
The **HTTPS** property must be set to `true` if **HTTP** and **SSLClient** have been set to `true`.

User response:
Edit the configuration file and define **HTTPS** as `true` for the given route.

MQCPE099 **<insert_1> on route <insert_2> requires MQ Advanced capabilities to be enabled**

Explanation:
A property specified for a route requires extended capabilities in IBM MQ Advanced. However, these capabilities are not enabled.

User response:
If you have IBM MQ Advanced, IBM MQ Advanced for z/OS, IBM MQ Advanced for z/OS VUE, or IBM MQ Appliance entitlement, enable the extended capabilities in IBM MQ Advanced with the **EnableAdvancedCapabilities** property. To use IBM MQ Advanced capabilities on a route, the local queue manager that is connected using the route is also required to have IBM MQ Advanced, IBM MQ Advanced for z/OS, IBM MQ Advanced for z/OS VUE, or IBM MQ Appliance entitlement.

MQCPE100 **Route <insert_1> site certificate label <insert_2> was not found in cryptographic hardware key store <insert_3>.**

Explanation:
A site certificate label was specified in the definition of the specified route, but a personal certificate with that label was not found in the cryptographic hardware key store.

User response:
Ensure that correct site certificate label is specified, and that a personal certificate with the label exists in the key store.

MQCPE101 **Invalid password protection mode specified.**

Explanation:
An invalid password protection mode was specified as a parameter to the **mqiptPW** command.

User response:
Rerun the **mqiptPW** command, specifying a valid protection mode.

MQCPE102 **Encryption key file cannot be specified with password protection mode <insert_1>.**

Explanation:
A password encryption key file was specified as a parameter to the **mqiptPW** command, but an encryption key cannot be used with the specified password protection mode.

User response:
Rerun the **mqiptPW** command, specifying a valid combination of parameters.

MQCPE103 **Encryption key file <insert_1> does not exist or cannot be read**

Explanation:
An encryption key file was specified for use by MQIPT or the **mqiptPW** command, but the file either does not exist or cannot be accessed.

User response:
Ensure that the encryption key file exists, the user that is running MQIPT or the **mqiptPW** command has read access to the file, and that the correct encryption key file is specified.

MQCPE104 **Error encrypting password <insert_1>**

Explanation:
An error occurred when encrypting a password.

User response:
Investigate the cause of the error in the exception that follows this message.

MQCPE105 **Error reading encryption key file <insert_1>**

Explanation:
An error occurred when reading the password encryption key file.

User response:
Ensure that the correct encryption key file is specified, and that the file is readable by the user running MQIPT or the **mqiptPW** command.

MQCPE106 **Error decrypting password in property <insert_1>**

Explanation:
The encrypted password in the specified property cannot be decrypted.

User response:
Ensure that the value of the property, or the contents of the file that the property references, is the output of running the **mqiptPW** command with a password protection mode specified that is supported by this version of MQIPT.

MQCPE107 **Error reading encrypted password file <insert_1>**

Explanation:
An error occurred when reading a file containing an encrypted password.

User response:
Ensure that the correct password file is specified, and that the file is readable by the user running MQIPT.

MQCPE108 **Property <insert_1> cannot be specified with property <insert_2>**

Explanation:
The two indicated properties cannot both be specified in the MQIPT configuration.

User response:
Edit the MQIPT configuration to specify only one of the indicated properties.

MQCPE109 **Encryption key file <insert_1> is not correctly formatted**

Explanation:
The contents of the specified password encryption key file is not in the correct format.

User response:
Ensure that the password encryption key file contains at least one character, and only one line of text.

MQCPE110 **MQIPT with name <insert_1> is already active**

Explanation:
MQIPT cannot start as there is an instance of MQIPT with the same name already active on the local machine.

User response:
Issue the command to start MQIPT, specifying a unique name for the instance of MQIPT to be started.

MQCPE111 **Local administration is unavailable**

Explanation:
Administration of local instances of MQIPT without using the command port is unavailable.

User response:
Ensure that the MQIPT installation is not corrupted, and that the Java runtime environment used when starting MQIPT or the **mqiptAdmin** command is the one supplied with MQIPT. If the problem persists, contact IBM Support.

MQCPE112 **Command port <insert_1> site certificate label <insert_2> was not found in key ring file <insert_3>.**

Explanation:

A site certificate label was specified for the command server listening on the specified port, but a personal certificate with that label was not found in the key ring file.

User response:
Ensure that correct site certificate label is specified, and that a personal certificate with the label exists in the appropriate key ring.

MQCPE113 **Command port <insert_1> site certificate label <insert_2> was not found in cryptographic hardware key store <insert_3>.**

Explanation:
A site certificate label was specified for the command server listening on the specified port, but a personal certificate with that label was not found in the cryptographic hardware key store.

User response:
Ensure that correct site certificate label is specified, and that a personal certificate with the label exists in the key store.

MQCPE114 **<insert_1> requires MQ Advanced capabilities to be enabled**

Explanation:
A property specified requires extended capabilities in IBM MQ Advanced. However, these capabilities are not enabled.

User response:
If you have IBM MQ Advanced, IBM MQ Advanced for z/OS, IBM MQ Advanced for z/OS VUE, or IBM MQ Appliance entitlement, enable the extended capabilities in IBM MQ Advanced with the **EnableAdvancedCapabilities** property.

MQCPE115 **The command server on port <insert_1> did not start because of the following error: <insert_2>**

Explanation:
The command server for one of the command ports did not start because of the specified error.

User response:
Correct the problem that is indicated in the error message. Then issue the refresh command to restart the command server.

MQCPE116 **Configuration errors detected during refresh**

Explanation:
MQIPT detected errors in the values of some configuration properties during the refresh process. The effective values for any properties affected by the errors have not been changed.

User response:
Correct the problems that are indicated in the preceding messages, then issue the refresh command again.

MQCPE117 **Invalid data received**

Explanation
An MQIPT route received invalid data that does not conform to the IBM MQ protocol.

This message is issued in the connection log to record connection activity.

User response:
Ensure that connections to the MQIPT route listener port are valid IBM MQ connections.

MQCPE118 **Transmission length <insert_1> exceeds the maximum transmission size <insert_2>**

Explanation
The indicated transmission length in data received by an MQIPT route exceeds the maximum transmission size for the route. This might indicate that the data is invalid or does not conform to the IBM MQ protocol.

This message is issued in the connection log to record connection activity.

User response:
Ensure that only IBM MQ clients and queue managers connect to the MQIPT route listener port.

MQCPE119 **The MQIPT service is not supported on this operating system**

Explanation:
The MQIPT service is not supported on this operating system because the operating system does not support System V init.

User response:
Use another method, such as `systemd` on Linux, to start MQIPT automatically.

MQCPE120 **The key ring <insert_1> could not be loaded because of the following error: <insert_2>**

Explanation:
A key ring file could not be loaded because of the specified error.

User response:
Ensure that the specified key ring is a PKCS #12 key ring, that user ID that started MQIPT has read access to the file, and that the MQIPT configuration specifies the correct password for the key ring.

MQCPE121 **The cryptographic hardware key store could not be loaded because of the following error: <insert_1>**

Explanation:
The PKCS #11 cryptographic hardware key store could not be loaded because of the specified error.

User response:
Ensure that the Java runtime environment (JRE) that runs MQIPT is correctly configured to access the cryptographic hardware key store, and that the MQIPT configuration specifies the correct password for the cryptographic hardware key store.

MQCPE122 **<insert_1> protocol not allowed.**

Explanation:
A connection is rejected as the route is not configured to accept the specified protocol.

User response:
Use the **AllowedProtocols** property to configure the protocols that are accepted by the route.

MQCPE123 **FIPS mode was requested but is unavailable.**

Explanation:
The request specified that the cryptographic operation must use a FIPS certified library. However, a FIPS certified library is not available.

User response:
Ensure that the Java Runtime Environment (JRE) that runs MQIPT is operating in FIPS mode, then retry the command.

MQCPE124 **The algorithm <insert_1> is not available while in FIPS mode.**

Explanation:
The protection mode specified is not valid for FIPS mode.

User response:
Change the algorithm specified or disable FIPS mode.

MQCPE125 **The value '<insert_1>' is not valid for the property '<insert_2>'. Valid values are: '<insert_3>'.**

Explanation:
An invalid property value has been specified.

User response:
Refer to IBM MQ Internet Pass-Thru configuration reference for full details of the valid values for each property.

MQCPE126 **Global property <insert 1> is invalid.**

Explanation:
A property specified in the global stanza of the configuration file is invalid for this version of MQIPT. The property is ignored.

User response:
Remove the invalid property from the MQIPT configuration file.

MQCPI001 **<insert_1> starting**

Explanation:
This MQIPT instance is beginning execution. Further initialization messages will follow.

MQCPI002 **<insert_1> shutting down**

Explanation:
MQIPT is going to shut down. This can result from a **STOP** command, or automatically if a configuration error prevents a successful startup or **REFRESH** action.

MQCPI003 **<insert_1> shutdown complete**

Explanation:
The shutdown process has completed. All MQIPT processes are now ended.

MQCPI004 **Reading configuration information from <insert_1>**

Explanation:
The MQIPT configuration file **mqipt.conf** is being read from the directory described in this message.

MQCPI005 **Listener port specified as not active - <insert_1> -> <insert_2>(<insert_3>)**

Explanation:
The route referred to in the message has been marked as inactive. No communication requests will be accepted on this route.

MQCPI006 **Route <insert_1> is starting and will forward messages to :**

Explanation:
A route has been started on the listener port shown in this message. This message is followed by other messages listing any properties associated with this route. Message MQCPI078 will be issued when the route is ready to accept connections.

MQCPI007 **Route <insert_1> has been stopped**

Explanation:
The route that was operating on the specified listener port is being shut down. This action normally occurs when a **REFRESH** command is issued to MQIPT and the route configuration has been changed.

MQCPI008 **Listening for control commands on port <insert_1> on local address <insert_2>**

Explanation:
This MQIPT instance is listening for control commands on the specified port and local address. An asterisk (*) indicates that MQIPT is listening for commands on all network interfaces.

MQCPI009 **Control command received: <insert_1>**

Explanation:
This message indicates that a control command has been received at the command port. Where applicable, details are included in the message.

MQCPI010 **Stopping command port on <insert_1>**

Explanation:
On a **REFRESH** operation, the command port is no longer in use in the new configuration. Commands will no longer be accepted at the specified port.

MQCPI011 **The path <insert_1> will be used to store the log files**

Explanation:
Logging output will be directed to the location described in this message, under the current configuration.

User response:
This may change if the configuration is amended and a **REFRESH** operation is requested.

MQCPI012 **Changing the value of MinConnectionThreads has no effect after the route is started**

Explanation:
The minimum number of connection threads is assigned at route startup and cannot be changed until MQIPT is restarted.

MQCPI013 **Connection from <insert_1> to host <insert_2> closed**

Explanation:
This message is issued in the connection log to record connection activity.

MQCPI014 **Protocol eyecatcher (<insert_1>) not recognized**

Explanation:
This message is issued in the connection log to record connection activity.

MQCPI015 **Client access has been disabled on this route**

Explanation:
This message is issued in the connection log to record connection activity.

MQCPI016 **Queue manager access has been disabled on this route**

Explanation:
This message is issued in the connection log to record connection activity.

MQCPI017 **A queue manager on <insert_1> was connected to host <insert_2>**

Explanation:
This message is issued in the connection log to record connection activity.

MQCPI018 **A client on <insert_1> was connected to host <insert_2>**

Explanation:
This message is issued in the connection log to record connection activity.

MQCPI019 **<insert_1> routes have been created - this exceeds the maximum number of supported routes, which is <insert_2>**

Explanation:
The maximum number of supported routes has been exceeded.

User response:
MQIPT will continue to operate, but you might want to create a second MQIPT instance and split the routes between the two.

MQCPI021 **Password checking has been enabled on the command port**

Explanation:
A password is required to access the command port.

MQCPI022 **Password checking has been disabled on the command port**

Explanation:
A password is not required to access the command port. To configure password authentication on the command port, set both the **RemoteCommandAuthentication** and **AccessPW** properties.

MQCPI024 **....and HTTP proxy at <insert_1>(<insert_2>)**

Explanation:
This message indicates that the outgoing connection for this route will be made using this HTTP proxy.

MQCPI025 **The refresh requested by Administration Client <insert_1> has finished**

Explanation:
As a result of receiving a **REFRESH** command, MQIPT is reading the configuration file and restarting.

MQCPI026 **Administration Client <insert_1> has requested shutdown**

Explanation:
As a result of receiving a **STOP** command, the MQIPT is shutting down.

MQCPI027 **<insert_1> command sent to MQIPT at <insert_2> on port <insert_3>**

Explanation:
The command specified has been sent by the administration client to MQIPT at the specified network address and command port.

MQCPI031 **.....cipher suites <insert_1>**

Explanation:
This message lists the cipher suites in use for this route or command port.

MQCPI032 **.....key ring file <insert_1>**

Explanation:
This message gives the file name of the key ring for this route or command port.

MQCPI033 **.....client authentication set to <insert_1>**

Explanation:
This message defines whether an SSL server is requesting client authentication for this route.

MQCPI034 **....<insert_1>(<insert_2>)**

Explanation:
This message shows the destination and destination port address for this route.

MQCPI035 **....using <insert_1> protocol**

Explanation:
This message shows the protocol being used to the route destination.

MQCPI036 **....SSL Client side enabled with properties :**

Explanation:
This message shows that the route will be using SSL/TLS to send data to the destination host.

MQCPI037 **....SSL Server side enabled with properties :**

Explanation:
This message shows that the route will be using SSL/TLS to receive data from the sending host.

MQCPI038 **.....peer certificate uses <insert_1>**

Explanation:
This message lists the distinguished names used to control authentication of peer certificates.

MQCPI039 **....and SOCKS proxy at <insert_1>(<insert_2>)**

Explanation:
This message shows that the outgoing connection for this route will be made using the specified SOCKS proxy.

MQCPI040 **Command port <insert_1> has been accessed by the Administration Client from network address <insert_2>**

Explanation:
The MQIPT command server listening on the specified port, has received a connection from the administration client at the specified remote network address.

MQCPI042 **Maximum connections reached on route <insert_1> - further requests will be blocked**

Explanation:
This message is written to the system console when the maximum number of connections has been reached for the given route. Further requests will be blocked until a connection becomes free or the **MaxConnectionThreads** value is increased.

MQCPI043 **Connections on route <insert_1> now unblocked**

Explanation:
This message is written to the system console when the given route is unblocked for connection requests.

MQCPI047 **.....CA key ring file <insert_1>**

Explanation:
This message gives the file name of the CA key ring for this route.

MQCPI048 **The ping by Administration Client <insert_1> has finished**

Explanation:
Response message from the `IPTController` to Administration Client.

MQCPI050 **Adding entry to inittab to automatically start MQIPT at system startup**

Explanation:
User has run the **mqiptService** script to start MQIPT as a system service.

MQCPI051 **Removing entry from inittab that automatically starts MQIPT at system startup**

Explanation:
User has run the **mqiptService** script to remove MQIPT from starting as a system service.

MQCPI052 **....Socks server side enabled**

Explanation:

This route will act as a SOCKS server (proxy) and will accept connections from a socksified application.

MQCPI057 **....trace level <insert_1> enabled**

Explanation:
This message is written to the system console when a route is started to show the level of tracing enabled on this route.

MQCPI058 **....and a URI name of <insert_1>**

Explanation:
This message is written to the system console when a route is started to show the Uniform Resource Identifier name on this route.

MQCPI060 **Installing files to automatically start MQIPT at system startup**

Explanation:
User has run the **mqiptService** script to start MQIPT as a system service.

MQCPI061 **Removing files that automatically starts MQIPT at system startup**

Explanation:
User has run the **mqiptService** script to remove MQIPT from starting as a system service.

MQCPI064 **......no SSL authentication on this route**

Explanation:
This message is written to the system console when a route is started and shows there is no SSL authentication is in use for this route, as an anonymous cipher suite has been specified.

MQCPI066 **....and HTTP server at <insert_1>(<insert_2>)**

Explanation:
This message indicates that the outgoing connection for this route will be made using this HTTP server.

MQCPI069 **....binding to local address <insert_1> when making new connections**

Explanation:
This message shows the local IP address each new connection is bound to the destination address. This should only be used on a multihomed system.

MQCPI070 **....using local port address range <insert_1>-<insert_2> when making new connections**

Explanation:
This message shows the local port addresses that will be used for new connections. This will allow firewall administrators to restrict connections from MQIPT.

MQCPI071 **......site certificate uses <insert_1>**

Explanation:
This message lists the distinguished names used to control selection of a site certificate.

MQCPI072 **......and certificate label <insert_1>**

Explanation:
This message lists the label name used to control selection of a site certificate.

MQCPI073 **Updated file <insert_1>**

Explanation:
The specified file has been updated by the **mqiptPW** command.

MQCPI074 **Created file <insert_1>**

Explanation:
The specified file has been created by the **mqiptPW** command.

MQCPI075 **....LDAP main server at <insert_1>(<insert_2>)**

Explanation:
This message lists the name of the main LDAP server used for CRL support.

MQCPI076 **....LDAP backup server at <insert_1>(<insert_2>)**

Explanation:
This message lists the name of the backup LDAP server used for CRL support.

MQCPI077 **....LDAP errors will be ignored**

Explanation:
This message means that any errors received from LDAP will be ignored.

MQCPI078 **Route <insert_1> ready for connection requests**

Explanation:
This message is displayed when a route is ready to accept connection requests.

MQCPI079 **....using security exit <insert_1>**

Explanation:
This message is written to the system console when a route is started to show the fully qualified name of the security exit.

MQCPI080 **......and timeout of <insert_1> second(s)**

Explanation:
This message is written to the system console when a route is started to show the timeout value of the security or certificate exit.

MQCPI083 **....refresh commands will not restart the route**

Explanation:
This message indicates that when a refresh command has been issued the route will not be restarted.

MQCPI084 **.....CRL cache expiry timeout is <insert_1> hour(s)**

Explanation:
This console message displays how long a CRL (or ARL) will remain in the MQIPT cache.

MQCPI085 **....CRLs will be saved in the key ring file(s)**

Explanation:
This console message means that any CRLs (or ARLs) retrieved from an LDAP server will be saved in the key ring file, attached to the associated CA certificate.

MQCPI086 **.....timeout of <insert_1> second(s)**

Explanation:
This message is written to the system console when a route is started to show the timeout value for connecting to the LDAP server.

MQCPI087 **.....userid is <insert_1>**

Explanation:
This message is written to the system console when a route is started to show the userid name to connect to the LDAP server.

MQCPI088 **....buffer size <insert_1>**

Explanation:
This message is written to the system console when a route is started to show the size of buffers being used, but only if not the value of 65535. This value will only be used if greater than the default value of 65535.

MQCPI090 **.....search baseDN uses <insert_1>**

Explanation:
This message is written to the system console when a route is started to show the LDAP baseDN key names to retrieve CRLs (and ARLs).

MQCPI091 **....allow plain connections**

Explanation:
This message is written to the system console when a route is started to indicate that plain connections are allowed when acting as an SSL server or running in SSL proxy mode.

MQCPI092 **....socket timeout <insert_1> ms**

Explanation:
This message shows the socket timeout value (in milliseconds)

MQCPI127 **....in full duplex mode**

Explanation:
This message shows the HTTP protocol being used to the destination is working in full duplex mode.

MQCPI128 **....in half duplex mode**

Explanation:
This message shows the HTTP protocol being used to the destination is working in half duplex mode.

MQCPI129 **.....using certificate exit <insert_1>**

Explanation:
This message is written to the system console when a route is started. Used to show the fully qualified name of the certificate exit.

MQCPI130 **Connection to caller closed due to connection failure to destination**

Explanation:
This message is written to the connection log for the closed connection to the caller, when MQIPT failed to connect to the target destination.

User response:
See previous connection failure for reason of closure.

MQCPI131 **.....and certificate exit data "<insert_1>"**

Explanation:
This message is written to the system console when a route is started. Used to show the data for the certificate exit.

MQCPI132 **....listening on local address <insert_1>**

Explanation:
This message shows the local IP address the route is listening on. This should only be used on a multihomed system.

MQCPI139 **.....secure socket protocols <insert_1>**

Explanation:
This message lists the secure socket protocol versions enabled for this route or command port.

MQCPI140 **....TCP keep alive enabled**

Explanation:
This message shows that TCP keep alive parameter has been enabled

MQCPI141 **.....cryptographic hardware key store**

Explanation:

This route or command port uses cryptographic hardware that supports the PKCS #11 interface for either the server or client key store.

MQCPI142 **.....cryptographic hardware CA key store**

Explanation:
This route uses cryptographic hardware that supports the PKCS #11 interface for either the server or client CA key store.

MQCPI143 **MQ Advanced capabilities enabled**

Explanation:
IBM MQ Advanced extended capabilities are enabled.

MQCPI144 **MQ Advanced capabilities not enabled**

Explanation:
IBM MQ Advanced extended capabilities are not enabled.

MQCPI145 **Enter password**

Explanation:
Prompt to enter a password by the **mqiptPW** command.

MQCPI150 **No password specified.**

Explanation:
No password was specified for the **mqiptPW** command to encrypt. The program terminates.

MQCPI151 **Reading password encryption key from <insert_1>**

Explanation:
The encryption key for passwords stored by MQIPT is being read from the specified file.

MQCPI152 **MQIPT name is <insert_1>**

Explanation:
The name for this instance of MQIPT is displayed.

MQCPI153 **Password checking is optional on the command port**

Explanation:
Connections to the command port can optionally supply a password for authentication. The password is checked if it is supplied.

MQCPI155 **Listening for control commands on port <insert_1> on local address <insert_2> using TLS**

Explanation:
This MQIPT instance is listening for control commands on the specified port and local address. Connections to this port are secured using TLS. An asterisk (*) indicates that MQIPT is listening for commands on all network interfaces.

MQCPI158 **....connection protocols accepted: <insert_1>**

Explanation:
The route accepts connections that use the specified protocols.

MQCPW001 **CRL expired for <insert_1>**

Explanation:
This message is displayed when a CRL (or ARL) is retrieved from an LDAP server.

User response:
Update the specified CRL in the LDAP server.

MQCPW003 **....Expired CRLs will be ignored**

Explanation:
This console message means that any expired CRLs (or ARLs) will be ignored and the connection request may be allowed.

MQCPW004 **.....Client certificate authentication is not set to 'required'. The certificate exit might not be called.**

Explanation

This console message is displayed when a route starts to indicate a conflict between the values of the **SSLServerExit** property and the **SSLServerClientAuth** or **SSLServerAskClientAuth** properties.

When the **SSLServerClientAuth** and **SSLServerAskClientAuth** properties in the route configuration indicate that client certificate authentication is optional or disabled, the TLS/SSL client is not required to present a certificate during the TLS/SSL handshake. If a certificate is not sent by the TLS/SSL client when it connects to MQIPT, the certificate exit is not called.

User response:
Use the **SSLServerClientAuth** or **SSLServerAskClientAuth** property to configure the route to require client certificate authentication if the certificate exit must be called for each connection.

MQCPW005 **Route <insert_1> <insert_2> key ring file <insert_3> certificate <insert_4> serial number <insert_5> is not yet valid. The certificate cannot be used before <insert_6>.**

Explanation:
This console message is displayed at route startup if one of the key ring files contains a certificate which is not yet valid because its Not Before date is in the future.

User response:
Check that the system clock is set correctly. If your organization operates its own CA, check the system clock on the CA system.

MQCPW006 **Route <insert_1> <insert_2> key ring file <insert_3> certificate <insert_4> serial number <insert_5> has expired. The certificate cannot be used after <insert_6>.**

Explanation:
This console message is displayed at route startup if one of the key ring files contains a certificate which has expired.

User response:
Check that the system clock is set correctly. If the clock is set correctly, obtain a replacement certificate.

MQCPW007 **Route <insert_1> property <insert_2> is invalid.**

Explanation:
A property specified for this route is invalid for this version of MQIPT. The property will be ignored and the route will continue to start up.

User response:
Remove the invalid property from the route definition.

MQCPW008 **Route <insert_1> certificate <insert_2> serial number <insert_3> is not yet valid. The certificate cannot be used before <insert_4>. The certificate is stored in the cryptographic hardware key store <insert_5>.**

Explanation:
This console message is displayed at route startup if the cryptographic hardware key store contains a certificate which is not yet valid because its Not Before date is in the future.

User response:
Check that the system clock is set correctly. If your organization operates its own CA, check the system clock on the CA system.

MQCPW009 **Route <insert_1> certificate <insert_2> serial number <insert_3> has expired. The certificate cannot be used after <insert_4>. The certificate is stored in the cryptographic hardware key store <insert_5>.**

Explanation:
This console message is displayed at route startup if the cryptographic hardware key store contains a certificate which has expired.

User response:
Check that the system clock is set correctly. If the clock is set correctly, obtain a replacement certificate.

MQCPW010 **Deprecated command syntax used.**

Explanation:
A command was issued using a syntax that is deprecated, and which does not offer the full range of command options. The deprecated syntax of the **mqiptPW** command does not allow passwords to be encrypted using the most secure method.

User response:
Review the command syntax and plan to issue the command using the latest syntax in the future.

MQCPW011 **Unprotected or weakly protected password specified in property <insert_1>**

Explanation:
A plain text or weakly protected password is specified in the indicated property.

User response:
To store the password securely, use the **mqiptPW** command to encrypt the password with the latest protection mode.

MQCPW012 **Unprotected or weakly protected password specified in property <insert_1> for route <insert_2>**

Explanation:
A plain text or weakly protected password is specified in the indicated property on the specified route.

User response:
To store the password securely, use the **mqiptPW** command to encrypt the password with the latest protection mode.

MQCPW013 **Command port <insert_1> is unprotected**

Explanation:
MQIPT is configured to listen for commands on the indicated port, but this port is not secured with TLS. Other systems on the network might be able to view data sent to this port by the **mqiptAdmin** command, including sensitive data such as the MQIPT access password.

User response:
Use the **SSLCommandPort** property to configure a command port that is secured with TLS.

MQCPW014 **Command port <insert_1> certificate <insert_2> serial number <insert_3> is not yet valid. The certificate cannot be used before <insert_4>. The certificate**

is stored in the key ring file <insert_5>.

Explanation:
The key ring file used by the command server for the specified command port contains a certificate which is not yet valid because its Not Before date is in the future.

User response:
Check that the system clock is set correctly. If your organization operates its own CA, check the system clock on the CA system.

MQCPW015 **Command port <insert_1> certificate <insert_2> serial number <insert_3> has expired. The certificate cannot be used after <insert_4>. The certificate is stored in the key ring file <insert_5>.**

Explanation:
The key ring file used by the command server for the specified command port contains a certificate which has expired.

User response:
Check that the system clock is set correctly. If the clock is set correctly, obtain a replacement certificate.

MQCPW016 **Command port <insert_1> certificate <insert_2> serial number <insert_3> is not yet valid. The certificate cannot be used before <insert_4>. The certificate is stored in the cryptographic hardware key store <insert_5>.**

Explanation:
The cryptographic hardware key store used by the command server for the specified command port contains a certificate which is not yet valid because its Not Before date is in the future.

User response:
Check that the system clock is set correctly. If your organization operates its own CA, check the system clock on the CA system.

MQCPW017 **Command port <insert_1> certificate <insert_2> serial number <insert_3> has expired. The certificate cannot be used after <insert_4>. The certificate is stored in the cryptographic hardware key store <insert_5>.**

Explanation:
The cryptographic hardware key store used by the command server for the specified command port contains a certificate which has expired.

User response:
Check that the system clock is set correctly. If the clock is set correctly, obtain a replacement certificate.

MQCAxxxx: administration client messages (CD)

MQCAE001 **Unknown host: <insert_1>**

Explanation:
The MQIPT host cannot be found.

User response:
Check the host name was specified correctly. Try to PING the host name or use its IP address.

MQCAE002 **The following error was reported by the system: <insert_1>**

Explanation:
An error has occurred while communicating with MQIPT.

User response:
Review the text of the error message and take the appropriate action.

MQCAE023 **MQIPT did not recognize your password**

Explanation:
An incorrect password was specified when issuing an administrative command to MQIPT.

User response:
Ensure that the correct password is entered when issuing administrative commands to MQIPT.

MQCAE024 **MQIPT <insert_1> has not recognized the command**

Explanation:
An error has been detected while communicating with the MQIPT.

User response:
Ensure that the **mqiptAdmin** command is at the same version or higher than the MQIPT instance being administered. If the condition persists, contact IBM Software Support.

MQCAE026 **Remote shutdown is disabled on MQIPT <insert_1>**

Explanation:
MQIPT has received a request to shut down but cannot proceed because remote shutdown is not enabled.

User response:
To enable remote shutdown of MQIPT, edit the configuration file and set the **RemoteShutDown** property to `true`.

MQCAE057 **Invalid port number specified**

Explanation:
An invalid port number was specified as a parameter to a command.

User response:
Specify a valid port number in the command parameters.

MQCAE058 **No local instance of MQIPT found with the name <insert_1>**

Explanation:
No active local instance of MQIPT with the specified name was found by the **mqiptAdmin** command.

User response:
Specify the name of an active instance of MQIPT as a parameter to the **mqiptAdmin** command. The instance of MQIPT must be running on the local machine under the same user ID as the **mqiptAdmin**. Alternatively, on AIX and Linux, **mqiptAdmin** can be run as `root`. To list active local instances of MQIPT, use the **-list** option of the **mqiptAdmin** command.

MQCAE059 **MQIPT <insert_1> is already shutting down**

Explanation:
A command was sent to MQIPT, but the MQIPT instance is already shutting down.

MQCAE060 **Local administration is disabled**

Explanation:
A command was sent to local instance of MQIPT, but local administration is disabled.

User response:
Either enable local administration using the **LocalAdmin** property, or administer MQIPT using the command port.

MQCAE061 **Error establishing TLS connection to MQIPT: <insert_1>**

Explanation:
An error occurred when the **mqiptAdmin** command tried to establish a TLS connection to MQIPT. Details of the exception that was thrown follow this message.

User response

Examine the details of the exception in the message. Ensure that the TLS configuration for the **mqiptAdmin** command is correct.

- The properties file must contain the name of the trust store to be used by **mqiptAdmin** to validate the identity of the remote MQIPT instance, and the password to access the trust store.
- The trust store used by **mqiptAdmin** must contain the the CA certificate of the CA that signed the MQIPT server certificate.

MQCAE062 **The file <insert_1> specified in the property <insert_2> does not exist or cannot be read.**

Explanation:
A property in the **mqiptAdmin** properties file refers to a file or directory which does not exist.

User response:
Edit the properties file and specify the correct location for the file or directory.

MQCAE063 **Properties file <insert_1> does not exist or cannot be read.**

Explanation:
The properties file specified with the **-p** parameter to **mqiptAdmin** does not exist or cannot be read.

User response:
Ensure that the name of the properties file is specified correctly with the **-p** parameter, and that the file can be read by the user that is running the **mqiptAdmin** command.

MQCAE064 **Error reading properties file <insert_1>.**

Explanation:
An error occurred when reading the properties file specified with the **-p** parameter to **mqiptAdmin**.

User response:
Ensure that the name of the properties file is specified correctly with the **-p** parameter, and that the file is can be read by the user that is running the **mqiptAdmin** command.

MQCAE065 **MQIPT <insert_1> has shut down.**

Explanation:
MQIPT has shut down while processing a command issued by **mqiptAdmin**.

MQCAE066 **MQIPT command level <insert_1> is not supported by mqiptAdmin version <insert_2>.**

Explanation:
The **mqiptAdmin** command is connected to an instance of MQIPT which is at a command level that is not supported by this version of **mqiptAdmin**.

User response:
Ensure that the **mqiptAdmin** command is always at the same version, or a higher version, than the instance of MQIPT it is connected to.

MQCAI025 **MQIPT <insert_1> has been refreshed**

Explanation:
MQIPT has been updated by reading its configuration file.

MQCAI026 **MQIPT <insert_1> has received shutdown request**

Explanation:
MQIPT has acknowledged receipt of a shutdown request and will now shut down.

MQCAI105 **Sending <insert_1> command to MQIPT instance with name <insert_2>**

Explanation:
mqiptAdmin is sending the specified command to the local instance of MQIPT with the specified name.

MQCAI106 **Local active instances of MQIPT:**

Explanation:
This message is displayed by the **mqiptAdmin** command to list the local active instances of MQIPT.

MQCAI107 **No local active instances of MQIPT**

Explanation:
The **mqiptAdmin** command found no local active instances of MQIPT running under the same user ID as the **mqiptAdmin** command.

MQCAI108 **If the command port is secured with TLS, you must specify the -s parameter to enable a secure connection.**

Explanation:
The **mqiptAdmin** command connected to MQIPT using an unsecured connection, but received a response that indicated that the command port is secured with TLS.

User response:
To connect to MQIPT using TLS, specify the **-s** parameter to the **mqiptAdmin** command.

MQCAI109 **The connection to MQIPT is secured with <insert_1>.**

Explanation:
The **mqiptAdmin** command has connected to MQIPT. The connection is secured using the indicated protocol.

MQCAW001 **The connection to MQIPT is unsecured.**

Explanation:
The **mqiptAdmin** command has connected to MQIPT using an unsecured connection.

User response:
None required. To connect to MQIPT securely using TLS, specify the **-s** parameter to the **mqiptAdmin** command.

MQCPxxxx: MQIPT messages (CD)

MQCPA104 **Command complete from MQIPT server at <insert_1>**

Explanation:
A command sent by the **mqiptAdmin** command has been accepted and run by MQIPT.

MQCPE001 **Directory does not exist or is not a directory <insert_1>**

Explanation:
At MQIPT initialization, a required directory could not be found. This message refers to a directory specified either in the MQIPT configuration file `mqipt.conf` or in the MQIPT command line startup options on the default directory.

User response:
Specify the correct directory and retry the command.

MQCPE004 **Route startup failed on port <insert_1>**

Explanation:
It was not possible to start the route with the specified **ListenerPort** number.

User response:
An I/O error occurred during route startup. Check for other adjacent error messages and log records to provide further explanation of the problem.

MQCPE005 **The configuration file <insert_1> could not be found**

Explanation:
The MQIPT configuration file `mqipt.conf` could not be found in the specified directory

User response:
Specify the correct directory and retry the command.

MQCPE006 **The number of routes has exceeded <insert_1>. MQIPT will start but this configuration is unsupported**

Explanation:
Your configuration has exceeded the maximum supported number of routes for one instance of MQIPT. Operation will not be halted but the system might become unstable or overloaded as a result. Configurations that exceed the stated maximum number of routes will not be supported.

User response:

Consider starting additional instances of MQIPT with fewer routes per instance.

MQCPE007 **Route not restarted on listener port <insert_1>**

Explanation:
During a refresh operation, the route that was operating on the specified **ListenerPort** was not restarted with the new configuration.

User response:
Check for other adjacent error messages for further explanation of the problem.

MQCPE008 **Duplicate route defined for listener port <insert_1>**

Explanation:
More than one route has been defined with the same **ListenerPort** value.

User response:
Remove the duplicate route from the configuration file and retry the command.

MQCPE009 **Log directory <insert_1> is not valid**

Explanation:
The indicated log directory either does not exist or is not accessible.

User response:
Check the directory exists and is accessible by MQIPT.

MQCPE010 **Listener or command port number <insert_1> is not valid**

Explanation:
The port address supplied for the command port or listener port parameter is invalid.

User response:
Specify a valid port address that is free for use. For guidance on use of port addresses in your network, consult your network administrator.

MQCPE012 **The value <insert_1> is not valid for the property <insert_2>**

Explanation:
An invalid property value has been specified.

User response:
Refer to IBM MQ Internet Pass-Thru configuration reference for full details of the valid values for each property.

MQCPE013 **ListenerPort property was not found in route <insert_1>**

Explanation:
MQIPT has detected a route in the configuration file that does not contain a **ListenerPort** property. The **ListenerPort** property is the primary and unique identifier for each route, and is therefore mandatory.

User response:
Specify a valid **ListenerPort** property for the given route.

MQCPE014 **ListenerPort property value <insert_1> is not valid**

Explanation:
An invalid port address is specified for the **ListenerPort** property of a route.

User response:
Specify a port number in the range 0 to 65535 for each **ListenerPort** property in the MQIPT configuration file.

MQCPE015 **No text was found for message number <insert_1>**

Explanation:
An internal error has been encountered for which no description is available.

User response:
The `mqipt.properties` file may have become corrupted and the specified message number could not be found. Check that the `mqipt.properties` file is in the `com.ibm.mq.ipt.jar` file, and that the message number is in the `mqipt.properties` file. If you are using the `MQIPT_PATH` environment variable, ensure it is set correctly.

MQCPE016 **The maximum number of connection threads is <insert_1> but this is less than the minimum number of connection threads, which is <insert_2>**

Explanation:
Your configuration file has specified the minimum number of connection threads with a value greater than the maximum number of connection threads.

User response:
This could be an error in a single route, a conflict between a global property and a route property, or a route property overriding the system default values. Refer to IBM MQ Internet Pass-Thru configuration reference for full details of the valid values and applicable defaults for each property.

MQCPE017 **The exception <insert_1> was thrown causing MQIPT to shut down**

Explanation:
MQIPT has abnormally terminated and has been shut down. This may have occurred because of system environmental conditions or constraints, such as memory overflow.

User response:
If the condition persists, contact IBM Software Support.

MQCPE018 **The ListenerPort property is blank - the route will not start**

Explanation:
The **ListenerPort** number has been omitted in a route.

User response:
Edit the configuration file and add a valid **ListenerPort**.

MQCPE019 **The stanza <insert_1> was not found before the following : <insert_2>**

Explanation:
A sequence error has occurred in the configuration file.

User response:
Edit the configuration file and make sure that all [route] stanzas are after the [global] stanza.

MQCPE020 **The new value for MaxConnectionThreads is <insert_1>. This must be greater than the current value <insert_2>**

Explanation:
After the route has started, the **MaxConnectionThread** property can only be increased.

User response:
Edit the configuration file and change the **MaxConnectionThread** property.

MQCPE021 **The Destination property was not supplied for route <insert_1>**

Explanation:
The **Destination** property is mandatory for a route, but was omitted in the route specified.

User response:
Edit the configuration file and add a **Destination** property for the given route.

MQCPE022 **The <insert_1> value <insert_2> is outside the valid range 1 - 65535**

Explanation:
The value of the **CommandPort** or **SSLCommandPort** property is outside the range 1-65535.

User response:
Edit the configuration file and change the value of the indicated property to a valid port number.

MQCPE023 **Request to shut down MQIPT from the Administration Client at <insert_1> is ignored because remote shutdown is disabled**

Explanation:
An attempt to shut down the MQIPT remotely has failed because remote shutdown was not enabled in the configuration file.

User response:
To enable remote shutdown of MQIPT, edit the configuration file and set the **RemoteShutDown** property to true.

MQCPE024 **The command received by the MQIPT controller has not been recognized**

Explanation:
The MQIPT command port received a command that it does not recognize.

User response:
Check the mqipt.log file for the identity of the command.

MQCPE025 **Failed to connect to server on host <insert_1>, port <insert_2>: <insert_3>**

Explanation:
The **mqiptAdmin** command could not connect to MQIPT at the indicated network address and port number.

User response:
Ensure that the MQIPT command port is configured to listen on the indicated port, and that MQIPT is running on indicated host. Check the MQIPT console log for any errors that occurred when starting the command server.

MQCPE026 **No reply received from server on host <insert_1>, port <insert_2>**

Explanation:
The **mqiptAdmin** command did not receive a reply from MQIPT. This can occur if the request times-out, or if there is a problem with MQIPT.

User response:
Check the MQIPT console log and connection log for additional error messages.

MQCPE027 **Reply from MQIPT not recognized**

Explanation:
The **mqiptAdmin** command received a reply from MQIPT which it did not recognize.

User response:
Ensure that the **mqiptAdmin** command is at the same version, or a higher version, than the instance of MQIPT it is connecting to.

MQCPE028 **Invalid stanza detected : <insert_1>**

Explanation:

The stated unrecognized stanza has been found in the configuration file.

User response:
Only `[global]` and `[route]` stanzas are valid in the configuration file.

MQCPE029 **Was not able to flush log output**

Explanation:
Some messages might not have been written to the log because the communication buffer could not be flushed.

User response:
Check that the MQIPT home directory file system is not full and that MQIPT still has access to write to the `logs` subdirectory.

MQCPE034 **Administration Client at <insert_1> did not supply the correct password**

Explanation:
The **AccessPW** property in the configuration file did not match that provided by the Administration Client.

User response:
Either change the **AccessPW** property in the configuration file or the saved password in the Administration Client.

MQCPE035 **Failed to start command listener on port <insert_1>**

Explanation:
An I/O error occurred starting the command port listener on the specified port.

User response:
Check the port number used for the **CommandPort** property in the configuration file.

MQCPE042 **There is a conflict with the following properties on route <insert_1> :**

Explanation:
Some properties can not be used with others. This message precedes the list of properties in conflict.

User response:
Check the following error messages and take the appropriate action.

MQCPE043 **....<insert_1> and <insert_2>**

Explanation:
The two specified properties cannot both be set at the same time on the same route.

User response:
Edit the configuration file and remove one of the specified properties on the given route.

MQCPE045 **....HTTP proxy or server name is missing**

Explanation:
The **HTTPProxy** or **HTTPServer** property must be set if the **HTTP** property has been set to `true`.

User response:
Edit the configuration file and define an **HTTPProxy** or **HTTPServer** for the given route.

MQCPE048 **Route startup failed on port <insert_1>, exception was : <insert_2>**

Explanation:
It was not possible to start the route with the specified **ListenerPort** number.

User response:
Check for other adjacent error messages and log records to provide further explanation of the problem.

MQCPE055 **....Socks proxy name is missing**

Explanation:
The **SocksProxy** property must be set if the **SocksClient** property has been set to true.

User response:
Edit the configuration file and define a **SocksProxy** for the given route.

MQCPE056 **Conflict with route properties**

Explanation:
Some properties cannot be used with others.

User response:
Check the console messages for details of the error and take the appropriate action.

MQCPE057 **SSL protocol (<insert_1>) was not recognized**

Explanation:
The route has been put into SSL/TLS proxy mode and the initial data flow is not recognized.

User response:
Make sure only SSL/TLS connections are being made to this route.

MQCPE058 **CONNECT request to <insert_3>(<insert_4>) through <insert_1>(<insert_2>) failed**

Explanation:
An HTTP CONNECT request was sent to the HTTP proxy to create an SSL tunnel to the HTTP server. The HTTP proxy did not send back a "200 OK" response to this request.

User response:

This can be caused by various problems. Enable tracing on the route and retry the connection. The trace file will show the real error.

MQCPE059 There are no defined key ring files

Explanation:
An SSL client or server has been defined without specifying at least one key ring file.

User response:
Use the **SSLClientKeyRing** and **SSLClientCAKeyRing** properties on the client side, or **SSLServerKeyRing** and **SSLServerCAKeyRing** on the server side, to define a key ring file and then restart the route.

MQCPE060 Runtime error setting SSL client connect timeout to <insert_1> seconds

Explanation:
An SSL runtime error has occurred on the client side setting the timeout value.

User response:
Check the value specified in the **SSLClientConnectTimeout** property is valid. Running a trace on the given route will show further error information.

MQCPE061 There are no enabled cipher suites

Explanation:
An SSL client or server connection has been started but MQIPT is unable to determine a valid cipher suite.

User response:
Check there are valid certificates in the defined key ring file(s). The private and public keys used to generate the certificates and the encryption algorithms used must comply with the list of supported cipher suites. See CipherSuites supported by MQIPT for the list of cipher suites supported by MQIPT.

MQCPE062 Runtime error setting SSL cipher suite <insert_1>

Explanation:
An unsupported SSL cipher suite has been defined on the client or server side.

User response:
Check the value specified in the **SSLClientCipherSuites** or **SSLServerCipherSuites** is valid and supported on this connection. Running a trace on the given route will show the list of enabled cipher suites. See CipherSuites supported by MQIPT for the list of cipher suites supported by MQIPT.

MQCPE063 File <insert_1> already exists - use the replace option

Explanation:
The file name parameter specified for the **mqiptPW** command already exists.

User response:
Either choose another file name or use the replace option.

MQCPE064 Runtime error generating decryption keys :\n <insert_1>

Explanation:
An error has occurred while generating cipher keys to decrypt the password used to open a key ring file.

User response:
The runtime error listed in the message should be rectified and the command run again.

MQCPE065 ....LDAP server name is missing

Explanation:
The **LDAPServer1** or **LDAPServer2** property must be set if the **LDAP** property has been set to `true`.

User response:
Edit the configuration file and define an **LDAPServer*** for the given route.

MQCPE066 ....LDAP password is missing for LDAP server <insert_1>

Explanation:
An LDAP userid has been specified without a password, for either the main or backup LDAP server.

User response:
Edit the configuration file and define an LDAP password for the given route. The **LDAPServer1Password** property is for the main server and **LDAPServer2Password** property is for the backup server.

MQCPE067 ....SSLClient or SSLServer missing for LDAP server

Explanation:
The **SSLClient** or **SSLServer** property must be set if the LDAP property has been set to true.

User response:
Edit the configuration file and define an **SSLClient** or **SSLServer** for the given route.

MQCPE068 ....Security exit name is missing

Explanation:
The **SecurityExitName** property must be set if the **SecurityExit** property has been set to `true`.

User response:
Edit the configuration file and define a **SecurityExitName** for the given route.

MQCPE071 Error writing to <insert_1>

Explanation:

An error occurred while creating or updating the file containing the encrypted password. The error message also contains the exception thrown.

User response:
This error is generated from the **mqiptPW** command. The error listed in the exception should be rectified and the command run again.

MQCPE072 **An unknown error occurred in security exit <insert_1>**

Explanation:
An error occurred in a user-defined security exit while validating a connection request.

User response:
Enable tracing in the security exit and try the connection request again. The error will be recorded in the security exit trace file.

MQCPE073 **Security exit <insert_1> timed out**

Explanation:
A user-defined security exit timed out while validating a connection request.

User response:
Increase the timeout period for the security exit and try the connection request again.

MQCPE074 **....Certificate exit name is missing**

Explanation:
The **SSLExitName** property must be set if the **SSLClientExit** or **SSLServerExit** property has been set to `true`.

User response:
Edit the configuration file and define a **SSLExitName** for the given route.

MQCPE075 **....SSLPlainConnections needs SSLServer or SSLProxyMode enabled**

Explanation:
The **SSLExitName** property must be set if the **SSLClientExit** or **SSLServerExit** property has been set to `true`.

User response:
Edit the configuration file and define a **SSLExitName** for the given route.

MQCPE076 **Route <insert_1> property <insert_2> contains unsupported CipherSuites. The following CipherSuites are unsupported: <insert_3>**

Explanation:
At least one unsupported cipher suite was included in the **SSLClientCipherSuites** or **SSLServerCipherSuites** property.

User response:
Edit the configuration file and remove the unsupported cipher suite from the route configuration.

MQCPE077 **Route <insert_1> property <insert_2> specifies file location <insert_3> which does not exist.**

Explanation:
A route property refers to a file or directory which does not exist.

User response:
Edit the configuration file and specify the correct location for the file or directory.

MQCPE078 **Route <insert_1> property <insert_2> specifies file location <insert_3> which cannot be read.**

Explanation:
A route property refers to a file that cannot be read.

User response:
Ensure that the file permissions allow MQIPT to read it.

MQCPE079 **Route <insert_1> site certificate label <insert_2> was not found in key ring file <insert_3>.**

Explanation:
A site certificate label was specified in the definition of the specified route, but a personal certificate with that label was not found in the key ring file.

User response:
Ensure that correct site certificate label is specified, and that a personal certificate with the label exists in the appropriate key ring.

MQCPE080 **Unable to determine MQIPT installation directory. Set the MQIPT_PATH environment variable to the absolute path of the top-level MQIPT directory.**

Explanation:
The MQIPT command was unable to determine the installation directory.

User response:
Set the `MQIPT_PATH` environment variable to the absolute path of the top-level MQIPT directory.

MQCPE081 **Invalid MQIPT_PATH <insert_1>. The directory does not exist or does not contain a valid MQIPT installation.**

Explanation:
The `MQIPT_PATH` environment variable is set incorrectly. Either the directory does not exist or the directory is not an MQIPT installation.

User response:
Check the MQIPT_PATH environment variable is set correctly and re-run the command.

MQCPE082 **Unable to install the MQIPT service because a service is already installed. Only one MQIPT service may be installed at a time.**

Explanation:
The user attempted to install the MQIPT service, but an MQIPT service is already installed. Only one MQIPT service may be installed on the system at a time.

User response:
Merge the required routes into the existing MQIPT service configuration, or remove the existing service and install the new service in its place.

MQCPE083 **Unable to remove the MQIPT service because the installed service was not installed by the current MQIPT installation. Run mqiptService from the MQIPT installation that installed the service.**

Explanation:
The MQIPT service can only be removed using the MQIPT installation that originally installed it. This error occurs when you have multiple MQIPT installations on the system and you attempt to remove the MQIPT service using a different installation from the one that originally installed it.

User response:
Run the command **mqiptService -remove** from the correct MQIPT installation.

MQCPE084 **The MQIPT service is not installed.**

Explanation:
The user attempted to remove the MQIPT service but there is no MQIPT service installed.

MQCPE086 **Security exit <insert_1> for route <insert_2> failed to initialize due to error <insert_3>.**

Explanation:
The security exit initialization method returned an unexpected error, which prevented the route from starting.

User response:
Investigate the cause of the error and restart the route.

MQCPE087 **Security exit <insert_1> for route <insert_2> failed to load due to error <insert_3>.**

Explanation:
The security exit could not be loaded, which prevented the route from starting.

User response:
Investigate the cause of the exit load error and restart the route.

MQCPE088 **Certificate exit <insert_1> for route <insert_2> failed to initialize due to error <insert_3>.**

Explanation:
The certificate exit initialization method returned an unexpected error, which prevented the route from starting.

User response:
Investigate the cause of the error and restart the route.

MQCPE089 **Certificate exit <insert_1> for route <insert_2> failed to load due to error <insert_3>.**

Explanation:
The certificate exit could not be loaded, which prevented the route from starting.

User response:
Investigate the cause of the exit load error and restart the route.

MQCPE090 **The security exit rejected the connection with return code <insert_1> and error <insert_2>.**

Explanation:
The security exit rejected a connection to the route listener port.

User response:
Investigate the error returned by the exit.

MQCPE091 **The SSLClient certificate exit rejected the connection with return code <insert_1> and error <insert_2>.**

Explanation:
The SSL client certificate exit rejected the remote server certificate.

User response:
Investigate the error returned by the exit.

MQCPE092 **The SSLServer certificate exit rejected the connection with return code <insert_1> and error <insert_2>.**

Explanation:
The SSL server certificate exit rejected the remote client certificate.

User response:
Investigate the error returned by the exit.

MQCPE093 **Global property <insert_1> specifies file location <insert_2> which does not exist.**

Explanation:
A global property refers to a file or directory which does not exist.

User response:
Edit the configuration file and specify the correct location for the file or directory.

MQCPE094 **Global property <insert_1> specifies file location <insert_2> which cannot be read.**

Explanation:
A global property refers to a file that cannot be read.

User response:
Ensure that the file permissions allow MQIPT to read it.

MQCPE095 **The MQIPT installation directory <insert_1> must not contain a space on this platform.**

Explanation:
The MQIPT installation directory contains a space character, which is not supported on AIX or Linux.

User response:
Rename the installation directory so that it does not contain a space.

MQCPE096 **Error enabling TCP keep alive**

Explanation:
The TCP keep alive route property is set, but MQIPT was unable to enable TCP keep alive.

User response:
Investigate the cause of the failure or disable TCP keep alive.

MQCPE097 **....SSLClient needs to be true for HTTPS communication**

Explanation:
The **SSLClient** property must be set to `true` if **HTTPS** has been set to `true`.

User response:
Edit the configuration file and define **SSLClient** as `true` for the given route.

MQCPE098 **....HTTPS needs to be true when SSLClient and HTTP are both set to true**

Explanation:
The **HTTPS** property must be set to `true` if **HTTP** and **SSLClient** have been set to `true`.

User response:
Edit the configuration file and define **HTTPS** as `true` for the given route.

MQCPE099 **<insert_1> on route <insert_2> requires MQ Advanced capabilities to be enabled**

Explanation:
A property specified for a route requires extended capabilities in IBM MQ Advanced. However, these capabilities are not enabled.

User response:
If you have IBM MQ Advanced, IBM MQ Advanced for z/OS, IBM MQ Advanced for z/OS VUE, or IBM MQ Appliance entitlement, enable the extended capabilities in IBM MQ Advanced with the **EnableAdvancedCapabilities** property. To use IBM MQ Advanced capabilities on a route, the local queue manager that is connected using the route is also required to have IBM MQ Advanced, IBM MQ Advanced for z/OS, IBM MQ Advanced for z/OS VUE, or IBM MQ Appliance entitlement.

MQCPE100 **Route <insert_1> site certificate label <insert_2> was not found in cryptographic hardware key store <insert_3>.**

Explanation:
A site certificate label was specified in the definition of the specified route, but a personal certificate with that label was not found in the cryptographic hardware key store.

User response:
Ensure that correct site certificate label is specified, and that a personal certificate with the label exists in the key store.

MQCPE101 **Invalid password protection mode specified.**

Explanation:
An invalid password protection mode was specified as a parameter to the **mqiptPW** command.

User response:
Rerun the **mqiptPW** command, specifying a valid protection mode.

MQCPE102 **Encryption key file cannot be specified with password protection mode <insert_1>.**

Explanation:
A password encryption key file was specified as a parameter to the **mqiptPW** command, but an encryption key cannot be used with the specified password protection mode.

User response:
Rerun the **mqiptPW** command, specifying a valid combination of parameters.

MQCPE103 **Encryption key file <insert_1> does not exist or cannot be read**

Explanation:
An encryption key file was specified for use by MQIPT or the **mqiptPW** command, but the file either does not exist or cannot be accessed.

User response:
Ensure that the encryption key file exists, the user that is running MQIPT or the **mqiptPW** command has read access to the file, and that the correct encryption key file is specified.

MQCPE104 **Error encrypting password <insert_1>**

Explanation:
An error occurred when encrypting a password.

User response:
Investigate the cause of the error in the exception that follows this message.

MQCPE105 **Error reading encryption key file <insert_1>**

Explanation:
An error occurred when reading the password encryption key file.

User response:
Ensure that the correct encryption key file is specified, and that the file is readable by the user running MQIPT or the **mqiptPW** command.

MQCPE106 **Error decrypting password in property <insert_1>**

Explanation:
The encrypted password in the specified property cannot be decrypted.

User response:
Ensure that the value of the property, or the contents of the file that the property references, is the output of running the **mqiptPW** command with a password protection mode specified that is supported by this version of MQIPT.

MQCPE107 **Error reading encrypted password file <insert_1>**

Explanation:
An error occurred when reading a file containing an encrypted password.

User response:
Ensure that the correct password file is specified, and that the file is readable by the user running MQIPT.

MQCPE108 **Property <insert_1> cannot be specified with property <insert_2>**

Explanation:
The two indicated properties cannot both be specified in the MQIPT configuration.

User response:
Edit the MQIPT configuration to specify only one of the indicated properties.

MQCPE109 **Encryption key file <insert_1> is not correctly formatted**

Explanation:
The contents of the specified password encryption key file is not in the correct format.

User response:
Ensure that the password encryption key file contains at least one character, and only one line of text.

MQCPE110 **MQIPT with name <insert_1> is already active**

Explanation:
MQIPT cannot start as there is an instance of MQIPT with the same name already active on the local machine.

User response:
Issue the command to start MQIPT, specifying a unique name for the instance of MQIPT to be started.

MQCPE111 **Local administration is unavailable**

Explanation:
Administration of local instances of MQIPT without using the command port is unavailable.

User response:
Ensure that the MQIPT installation is not corrupted, and that the Java runtime environment used when starting MQIPT or the **mqiptAdmin** command is the one supplied with MQIPT. If the problem persists, contact IBM Support.

MQCPE112 **Command port <insert_1> site certificate label <insert_2> was not found in key ring file <insert_3>.**

Explanation:
A site certificate label was specified for the command server listening on the specified port, but a personal certificate with that label was not found in the key ring file.

User response:
Ensure that correct site certificate label is specified, and that a personal certificate with the label exists in the appropriate key ring.

MQCPE113 **Command port <insert_1> site certificate label <insert_2> was not found in cryptographic hardware key store <insert_3>.**

Explanation:

A site certificate label was specified for the command server listening on the specified port, but a personal certificate with that label was not found in the cryptographic hardware key store.

User response:
Ensure that correct site certificate label is specified, and that a personal certificate with the label exists in the key store.

MQCPE114 **<insert_1> requires MQ Advanced capabilities to be enabled**

Explanation:
A property specified requires extended capabilities in IBM MQ Advanced. However, these capabilities are not enabled.

User response:
If you have IBM MQ Advanced, IBM MQ Advanced for z/OS, IBM MQ Advanced for z/OS VUE, or IBM MQ Appliance entitlement, enable the extended capabilities in IBM MQ Advanced with the **EnableAdvancedCapabilities** property.

MQCPE115 **The command server on port <insert_1> did not start because of the following error: <insert_2>**

Explanation:
The command server for one of the command ports did not start because of the specified error.

User response:
Correct the problem that is indicated in the error message. Then issue the refresh command to restart the command server.

MQCPE116 **Configuration errors detected during refresh**

Explanation:
MQIPT detected errors in the values of some configuration properties during the refresh process. The effective values for any properties affected by the errors have not been changed.

User response:
Correct the problems that are indicated in the preceding messages, then issue the refresh command again.

MQCPE117 **Invalid data received**

Explanation
An MQIPT route received invalid data that does not conform to the IBM MQ protocol.

This message is issued in the connection log to record connection activity.

User response:
Ensure that connections to the MQIPT route listener port are valid IBM MQ connections.

MQCPE118 **Transmission length <insert_1> exceeds the maximum transmission size <insert_2>**

Explanation
The indicated transmission length in data received by an MQIPT route exceeds the maximum transmission size for the route. This might indicate that the data is invalid or does not conform to the IBM MQ protocol.

This message is issued in the connection log to record connection activity.

User response:
Ensure that only IBM MQ clients and queue managers connect to the MQIPT route listener port.

MQCPE119 **The MQIPT service is not supported on this operating system**

Explanation:
The MQIPT service is not supported on this operating system because the operating system does not support System V init.

User response:
Use another method, such as `systemd` on Linux, to start MQIPT automatically.

MQCPE120 **The key ring <insert_1> could not be loaded because of the following error: <insert_2>**

Explanation:
A key ring file could not be loaded because of the specified error.

User response:
Ensure that the specified key ring is a PKCS #12 key ring, that user ID that started MQIPT has read access to the file, and that the MQIPT configuration specifies the correct password for the key ring.

MQCPE121 **The cryptographic hardware key store could not be loaded because of the following error: <insert_1>**

Explanation:
The PKCS #11 cryptographic hardware key store could not be loaded because of the specified error.

User response:
Ensure that the Java runtime environment (JRE) that runs MQIPT is correctly configured to access the cryptographic hardware key store, and that the MQIPT configuration specifies the correct password for the cryptographic hardware key store.

MQCPE122 **<insert_1> protocol not allowed.**

Explanation:

A connection is rejected as the route is not configured to accept the specified protocol.

User response:
Use the **AllowedProtocols** property to configure the protocols that are accepted by the route.

MQCPE123 **FIPS mode was requested but is unavailable.**

Explanation:
The request specified that the cryptographic operation must use a FIPS certified library. However, a FIPS certified library is not available.

User response:
Ensure that the Java Runtime Environment (JRE) that runs MQIPT is operating in FIPS mode, then retry the command.

MQCPE124 **The algorithm <insert_1> is not available while in FIPS mode.**

Explanation:
The protection mode specified is not valid for FIPS mode.

User response:
Change the algorithm specified or disable FIPS mode.

MQCPE125 **The value '<insert_1>' is not valid for the property '<insert_2>'. Valid values are: '<insert_3>'.**

Explanation:
An invalid property value has been specified.

User response:
Refer to IBM MQ Internet Pass-Thru configuration reference for full details of the valid values for each property.

MQCPE126 **Global property <insert 1> is invalid.**

Explanation:
A property specified in the global stanza of the configuration file is invalid for this version of MQIPT. The property is ignored.

User response:
Remove the invalid property from the MQIPT configuration file.

MQCPI001 **<insert_1> starting**

Explanation:
This MQIPT instance is beginning execution. Further initialization messages will follow.

MQCPI002 **<insert_1> shutting down**

Explanation:
MQIPT is going to shut down. This can result from a **STOP** command, or automatically if a configuration error prevents a successful startup or **REFRESH** action.

MQCPI003 **<insert_1> shutdown complete**

Explanation:
The shutdown process has completed. All MQIPT processes are now ended.

MQCPI004 **Reading configuration information from <insert_1>**

Explanation:
The MQIPT configuration file **mqipt.conf** is being read from the directory described in this message.

MQCPI005 **Listener port specified as not active - <insert_1> -> <insert_2>(<insert_3>)**

Explanation:
The route referred to in the message has been marked as inactive. No communication requests will be accepted on this route.

MQCPI006 **Route <insert_1> is starting and will forward messages to :**

Explanation:
A route has been started on the listener port shown in this message. This message is followed by other messages listing any properties associated with this route. Message MQCPI078 will be issued when the route is ready to accept connections.

MQCPI007 **Route <insert_1> has been stopped**

Explanation:
The route that was operating on the specified listener port is being shut down. This action normally occurs when a **REFRESH** command is issued to MQIPT and the route configuration has been changed.

MQCPI008 **Listening for control commands on port <insert_1> on local address <insert_2>**

Explanation:
This MQIPT instance is listening for control commands on the specified port and local address. An asterisk (*) indicates that MQIPT is listening for commands on all network interfaces.

MQCPI009 **Control command received: <insert_1>**

Explanation:
This message indicates that a control command has been received at the command port. Where applicable, details are included in the message.

MQCPI010 **Stopping command port on <insert_1>**

Explanation:

On a **REFRESH** operation, the command port is no longer in use in the new configuration. Commands will no longer be accepted at the specified port.

MQCPI011 **The path <insert_1> will be used to store the log files**

Explanation:
Logging output will be directed to the location described in this message, under the current configuration.

User response:
This may change if the configuration is amended and a **REFRESH** operation is requested.

MQCPI012 **Changing the value of MinConnectionThreads has no effect after the route is started**

Explanation:
The minimum number of connection threads is assigned at route startup and cannot be changed until MQIPT is restarted.

MQCPI013 **Connection from <insert_1> to host <insert_2> closed**

Explanation:
This message is issued in the connection log to record connection activity.

MQCPI014 **Protocol eyecatcher (<insert_1>) not recognized**

Explanation:
This message is issued in the connection log to record connection activity.

MQCPI015 **Client access has been disabled on this route**

Explanation:
This message is issued in the connection log to record connection activity.

MQCPI016 **Queue manager access has been disabled on this route**

Explanation:
This message is issued in the connection log to record connection activity.

MQCPI017 **A queue manager on <insert_1> was connected to host <insert_2>**

Explanation:
This message is issued in the connection log to record connection activity.

MQCPI018 **A client on <insert_1> was connected to host <insert_2>**

Explanation:
This message is issued in the connection log to record connection activity.

MQCPI019 **<insert_1> routes have been created - this exceeds the maximum number of supported routes, which is <insert_2>**

Explanation:
The maximum number of supported routes has been exceeded.

User response:
MQIPT will continue to operate, but you might want to create a second MQIPT instance and split the routes between the two.

MQCPI021 **Password checking has been enabled on the command port**

Explanation:
A password is required to access the command port.

MQCPI022 **Password checking has been disabled on the command port**

Explanation:
A password is not required to access the command port. To configure password authentication on the command port, set both the **RemoteCommandAuthentication** and **AccessPW** properties.

MQCPI024 **....and HTTP proxy at <insert_1>(<insert_2>)**

Explanation:
This message indicates that the outgoing connection for this route will be made using this HTTP proxy.

MQCPI025 **The refresh requested by Administration Client <insert_1> has finished**

Explanation:
As a result of receiving a **REFRESH** command, MQIPT is reading the configuration file and restarting.

MQCPI026 **Administration Client <insert_1> has requested shutdown**

Explanation:
As a result of receiving a **STOP** command, the MQIPT is shutting down.

MQCPI027 **<insert_1> command sent to MQIPT at <insert_2> on port <insert_3>**

Explanation:
The command specified has been sent by the administration client to MQIPT at the specified network address and command port.

MQCPI031 **.....cipher suites <insert_1>**

Explanation:

This message lists the cipher suites in use for this route or command port.

MQCPI032 **.....key ring file <insert_1>**

Explanation:
This message gives the file name of the key ring for this route or command port.

MQCPI033 **.....client authentication set to <insert_1>**

Explanation:
This message defines whether an SSL server is requesting client authentication for this route.

MQCPI034 **....<insert_1>(<insert_2>)**

Explanation:
This message shows the destination and destination port address for this route.

MQCPI035 **....using <insert_1> protocol**

Explanation:
This message shows the protocol being used to the route destination.

MQCPI036 **....SSL Client side enabled with properties :**

Explanation:
This message shows that the route will be using SSL/TLS to send data to the destination host.

MQCPI037 **....SSL Server side enabled with properties :**

Explanation:
This message shows that the route will be using SSL/TLS to receive data from the sending host.

MQCPI038 **.....peer certificate uses <insert_1>**

Explanation:
This message lists the distinguished names used to control authentication of peer certificates.

MQCPI039 **....and SOCKS proxy at <insert_1>(<insert_2>)**

Explanation:
This message shows that the outgoing connection for this route will be made using the specified SOCKS proxy.

MQCPI040 **Command port <insert_1> has been accessed by the Administration Client from network address <insert_2>**

Explanation:
The MQIPT command server listening on the specified port, has received a connection from the administration client at the specified remote network address.

MQCPI042 **Maximum connections reached on route <insert_1> - further requests will be blocked**

Explanation:
This message is written to the system console when the maximum number of connections has been reached for the given route. Further requests will be blocked until a connection becomes free or the **MaxConnectionThreads** value is increased.

MQCPI043 **Connections on route <insert_1> now unblocked**

Explanation:
This message is written to the system console when the given route is unblocked for connection requests.

MQCPI047 **.....CA key ring file <insert_1>**

Explanation:
This message gives the file name of the CA key ring for this route.

MQCPI048 **The ping by Administration Client <insert_1> has finished**

Explanation:
Response message from the `IPTController` to Administration Client.

MQCPI050 **Adding entry to inittab to automatically start MQIPT at system startup**

Explanation:
User has run the **mqiptService** script to start MQIPT as a system service.

MQCPI051 **Removing entry from inittab that automatically starts MQIPT at system startup**

Explanation:
User has run the **mqiptService** script to remove MQIPT from starting as a system service.

MQCPI052 **....Socks server side enabled**

Explanation:
This route will act as a SOCKS server (proxy) and will accept connections from a socksified application.

MQCPI057 **....trace level <insert_1> enabled**

Explanation:
This message is written to the system console when a route is started to show the level of tracing enabled on this route.

MQCPI058 **....and a URI name of <insert_1>**

Explanation:
This message is written to the system console when a route is started to show the Uniform Resource Identifier name on this route.

MQCPI060 **Installing files to automatically start MQIPT at system startup**

Explanation:
User has run the **mqiptService** script to start MQIPT as a system service.

MQCPI061 **Removing files that automatically starts MQIPT at system startup**

Explanation:
User has run the **mqiptService** script to remove MQIPT from starting as a system service.

MQCPI064 **.....no SSL authentication on this route**

Explanation:
This message is written to the system console when a route is started and shows there is no SSL authentication is in use for this route, as an anonymous cipher suite has been specified.

MQCPI066 **....and HTTP server at <insert_1>(<insert_2>)**

Explanation:
This message indicates that the outgoing connection for this route will be made using this HTTP server.

MQCPI069 **....binding to local address <insert_1> when making new connections**

Explanation:
This message shows the local IP address each new connection is bound to the destination address. This should only be used on a multihomed system.

MQCPI070 **....using local port address range <insert_1>-<insert_2> when making new connections**

Explanation:
This message shows the local port addresses that will be used for new connections. This will allow firewall administrators to restrict connections from MQIPT.

MQCPI071 **.....site certificate uses <insert_1>**

Explanation:
This message lists the distinguished names used to control selection of a site certificate.

MQCPI072 **.....and certificate label <insert_1>**

Explanation:
This message lists the label name used to control selection of a site certificate.

MQCPI073 **Updated file <insert_1>**

Explanation:
The specified file has been updated by the **mqiptPW** command.

MQCPI074 **Created file <insert_1>**

Explanation:
The specified file has been created by the **mqiptPW** command.

MQCPI075 **....LDAP main server at <insert_1>(<insert_2>)**

Explanation:
This message lists the name of the main LDAP server used for CRL support.

MQCPI076 **....LDAP backup server at <insert_1>(<insert_2>)**

Explanation:
This message lists the name of the backup LDAP server used for CRL support.

MQCPI077 **....LDAP errors will be ignored**

Explanation:
This message means that any errors received from LDAP will be ignored.

MQCPI078 **Route <insert_1> ready for connection requests**

Explanation:
This message is displayed when a route is ready to accept connection requests.

MQCPI079 **....using security exit <insert_1>**

Explanation:
This message is written to the system console when a route is started to show the fully qualified name of the security exit.

MQCPI080 **.....and timeout of <insert_1> second(s)**

Explanation:
This message is written to the system console when a route is started to show the timeout value of the security or certificate exit.

MQCPI083 **....refresh commands will not restart the route**

Explanation:
This message indicates that when a refresh command has been issued the route will not be restarted.

MQCPI084 **.....CRL cache expiry timeout is <insert_1> hour(s)**

Explanation:
This console message displays how long a CRL (or ARL) will remain in the MQIPT cache.

MQCPI085 **....CRLs will be saved in the key ring file(s)**

Explanation:
This console message means that any CRLs (or ARLs) retrieved from an LDAP server will be saved in the key ring file, attached to the associated CA certificate.

MQCPI086 **.....timeout of <insert_1> second(s)**

Explanation:
This message is written to the system console when a route is started to show the timeout value for connecting to the LDAP server.

MQCPI087 **.....userid is <insert_1>**

Explanation:
This message is written to the system console when a route is started to show the userid name to connect to the LDAP server.

MQCPI088 **....buffer size <insert_1>**

Explanation:
This message is written to the system console when a route is started to show the size of buffers being used, but only if not the value of 65535. This value will only be used if greater than the default value of 65535.

MQCPI090 **.....search baseDN uses <insert_1>**

Explanation:
This message is written to the system console when a route is started to show the LDAP baseDN key names to retrieve CRLs (and ARLs).

MQCPI091 **....allow plain connections**

Explanation:
This message is written to the system console when a route is started to indicate that plain connections are allowed when acting as an SSL server or running in SSL proxy mode.

MQCPI092 **....socket timeout <insert_1> ms**

Explanation:
This message shows the socket timeout value (in milliseconds)

MQCPI127 **....in full duplex mode**

Explanation:
This message shows the HTTP protocol being used to the destination is working in full duplex mode.

MQCPI128 **....in half duplex mode**

Explanation:
This message shows the HTTP protocol being used to the destination is working in half duplex mode.

MQCPI129 **.....using certificate exit <insert_1>**

Explanation:
This message is written to the system console when a route is started. Used to show the fully qualified name of the certificate exit.

MQCPI130 **Connection to caller closed due to connection failure to destination**

Explanation:
This message is written to the connection log for the closed connection to the caller, when MQIPT failed to connect to the target destination.

User response:
See previous connection failure for reason of closure.

MQCPI131 **.....and certificate exit data "<insert_1>"**

Explanation:
This message is written to the system console when a route is started. Used to show the data for the certificate exit.

MQCPI132 **....listening on local address <insert_1>**

Explanation:
This message shows the local IP address the route is listening on. This should only be used on a multihomed system.

MQCPI139 **.....secure socket protocols <insert_1>**

Explanation:
This message lists the secure socket protocol versions enabled for this route or command port.

MQCPI140 **....TCP keep alive enabled**

Explanation:
This message shows that TCP keep alive parameter has been enabled

MQCPI141 **.....cryptographic hardware key store**

Explanation:
This route or command port uses cryptographic hardware that supports the PKCS #11 interface for either the server or client key store.

MQCPI142 **.....cryptographic hardware CA key store**

Explanation:
This route uses cryptographic hardware that supports the PKCS #11 interface for either the server or client CA key store.

MQCPI143 **MQ Advanced capabilities enabled**

Explanation:
IBM MQ Advanced extended capabilities are enabled.

MQCPI144 **MQ Advanced capabilities not enabled**

Explanation:
IBM MQ Advanced extended capabilities are not enabled.

MQCPI145 Enter password

Explanation:
Prompt to enter a password by the **mqiptPW** command.

MQCPI150 No password specified.

Explanation:
No password was specified for the **mqiptPW** command to encrypt. The program terminates.

MQCPI151 Reading password encryption key from <insert_1>

Explanation:
The encryption key for passwords stored by MQIPT is being read from the specified file.

MQCPI152 MQIPT name is <insert_1>

Explanation:
The name for this instance of MQIPT is displayed.

MQCPI153 Password checking is optional on the command port

Explanation:
Connections to the command port can optionally supply a password for authentication. The password is checked if it is supplied.

MQCPI155 Listening for control commands on port <insert_1> on local address <insert_2> using TLS

Explanation:
This MQIPT instance is listening for control commands on the specified port and local address. Connections to this port are secured using TLS. An asterisk (*) indicates that MQIPT is listening for commands on all network interfaces.

MQCPI158 ....connection protocols accepted: <insert_1>

Explanation:
The route accepts connections that use the specified protocols.

MQCPW001 CRL expired for <insert_1>

Explanation:
This message is displayed when a CRL (or ARL) is retrieved from an LDAP server.

User response:
Update the specified CRL in the LDAP server.

MQCPW003 ....Expired CRLs will be ignored

Explanation:
This console message means that any expired CRLs (or ARLs) will be ignored and the connection request may be allowed.

MQCPW004 ......Client certificate authentication is not set to 'required'. The certificate exit might not be called.

Explanation

This console message is displayed when a route starts to indicate a conflict between the values of the **SSLServerExit** property and the **SSLServerClientAuth** or **SSLServerAskClientAuth** properties.

When the **SSLServerClientAuth** and **SSLServerAskClientAuth** properties in the route configuration indicate that client certificate authentication is optional or disabled, the TLS/SSL client is not required to present a certificate during the TLS/SSL handshake. If a certificate is not sent by the TLS/SSL client when it connects to MQIPT, the certificate exit is not called.

User response:
Use the **SSLServerClientAuth** or **SSLServerAskClientAuth** property to configure the route to require client certificate authentication if the certificate exit must be called for each connection.

MQCPW005 Route <insert_1> <insert_2> key ring file <insert_3> certificate <insert_4> serial number <insert_5> is not yet valid. The certificate cannot be used before <insert_6>.

Explanation:
This console message is displayed at route startup if one of the key ring files contains a certificate which is not yet valid because its Not Before date is in the future.

User response:
Check that the system clock is set correctly. If your organization operates its own CA, check the system clock on the CA system.

MQCPW006 Route <insert_1> <insert_2> key ring file <insert_3> certificate <insert_4> serial number <insert_5> has expired. The certificate cannot be used after <insert_6>.

Explanation:
This console message is displayed at route startup if one of the key ring files contains a certificate which has expired.

User response:

Check that the system clock is set correctly. If the clock is set correctly, obtain a replacement certificate.

MQCPW007 **Route <insert_1> property <insert_2> is invalid.**

Explanation:
A property specified for this route is invalid for this version of MQIPT. The property will be ignored and the route will continue to start up.

User response:
Remove the invalid property from the route definition.

MQCPW008 **Route <insert_1> certificate <insert_2> serial number <insert_3> is not yet valid. The certificate cannot be used before <insert_4>. The certificate is stored in the cryptographic hardware key store <insert_5>.**

Explanation:
This console message is displayed at route startup if the cryptographic hardware key store contains a certificate which is not yet valid because its Not Before date is in the future.

User response:
Check that the system clock is set correctly. If your organization operates its own CA, check the system clock on the CA system.

MQCPW009 **Route <insert_1> certificate <insert_2> serial number <insert_3> has expired. The certificate cannot be used after <insert_4>. The certificate is stored in the cryptographic hardware key store <insert_5>.**

Explanation:
This console message is displayed at route startup if the cryptographic hardware key store contains a certificate which has expired.

User response:
Check that the system clock is set correctly. If the clock is set correctly, obtain a replacement certificate.

MQCPW010 **Deprecated command syntax used.**

Explanation:
A command was issued using a syntax that is deprecated, and which does not offer the full range of command options. The deprecated syntax of the **mqiptPW** command does not allow passwords to be encrypted using the most secure method.

User response:
Review the command syntax and plan to issue the command using the latest syntax in the future.

MQCPW011 **Unprotected or weakly protected password specified in property <insert_1>**

Explanation:
A plain text or weakly protected password is specified in the indicated property.

User response:
To store the password securely, use the **mqiptPW** command to encrypt the password with the latest protection mode.

MQCPW012 **Unprotected or weakly protected password specified in property <insert_1> for route <insert_2>**

Explanation:
A plain text or weakly protected password is specified in the indicated property on the specified route.

User response:
To store the password securely, use the **mqiptPW** command to encrypt the password with the latest protection mode.

MQCPW013 **Command port <insert_1> is unprotected**

Explanation:
MQIPT is configured to listen for commands on the indicated port, but this port is not secured with TLS. Other systems on the network might be able to view data sent to this port by the **mqiptAdmin** command, including sensitive data such as the MQIPT access password.

User response:
Use the **SSLCommandPort** property to configure a command port that is secured with TLS.

MQCPW014 **Command port <insert_1> certificate <insert_2> serial number <insert_3> is not yet valid. The certificate cannot be used before <insert_4>. The certificate is stored in the key ring file <insert_5>.**

Explanation:
The key ring file used by the command server for the specified command port contains a certificate which is not yet valid because its Not Before date is in the future.

User response:
Check that the system clock is set correctly. If your organization operates its own CA, check the system clock on the CA system.

MQCPW015 **Command port <insert_1> certificate <insert_2> serial number <insert_3> has expired.**

The certificate cannot be used after <insert_4>. The certificate is stored in the key ring file <insert_5>.

Explanation:
The key ring file used by the command server for the specified command port contains a certificate which has expired.

User response:
Check that the system clock is set correctly. If the clock is set correctly, obtain a replacement certificate.

MQCPW016 **Command port <insert_1> certificate <insert_2> serial number <insert_3> is not yet valid. The certificate cannot be used before <insert_4>. The certificate is stored in the cryptographic hardware key store <insert_5>.**

Explanation:
The cryptographic hardware key store used by the command server for the specified command port contains a certificate which is not yet valid because its Not Before date is in the future.

User response:
Check that the system clock is set correctly. If your organization operates its own CA, check the system clock on the CA system.

MQCPW017 **Command port <insert_1> certificate <insert_2> serial number <insert_3> has expired. The certificate cannot be used after <insert_4>. The certificate is stored in the cryptographic hardware key store <insert_5>.**

Explanation:
The cryptographic hardware key store used by the command server for the specified command port contains a certificate which has expired.

User response:
Check that the system clock is set correctly. If the clock is set correctly, obtain a replacement certificate.

JSON format diagnostic messages

A tabular description of the name/value pairs that make up each JSON format IBM MQ diagnostic message.

For more information on diagnostic messages, see QMErrorLog service.

If you write an error log file in JSON format, each error message contains single lines of JSON.

Table 10. Name/value pairs in the message object

name	Type	Description
host	string	The host name
ibm_arithInsert1	number	The first message variable.
ibm_arithInsert2	number	The second message variable.
ibm_commentInsert1	string	The third message variable, if required.
ibm_commentInsert2	string	The fourth message variable, if required.
ibm_commentInsert3	string	The fifth message variable, if required.
ibm_datetime	string	An ISO 8601 formatted timestamp indicating when the message was generated. Of the form `YYYY-MM-DDTHH:MM:SS.mmmZ`, always in UTC.
ibm_installationDir	string	The installation path. Included because it allows a parsing program on the machine to run appropriate commands from the installation.
ibm_installationName	string	The installation name.
ibm_messageID	string	The diagnostic message identifier including the severity character, for example, AMQ6209W.
ibm_processID	number	The process identifier.

Table 10. Name/value pairs in the message object (continued)		
name	**Type**	**Description**
ibm_processName	string	The process, or job name on IBM i, for example, `amqzxma0`.
ibm_qmgrId	string	An identifier for the queue manager.
ibm_remoteHost	string	IP address of the associated client program, if there is one.
ibm_sequence	string	Sequence number of message; intended to differentiate between messages produced at the same time.
ibm_serverName	string	The name of the queue manager.
ibm_threadId	number	The IBM MQ thread identifier within the process.
ibm_userName	string	The real name of the user under which the process is running.
ibm_version	string	IBM MQ Version, Release, Modification, Fix pack (VRMF) information.
loglevel	string	Either, INFO, WARNING, or ERROR.
message	string	A summary of the message, including the identifier, with inserts expanded.
module	string	The source file and line number where the message was generated, for example, `amqxerrx.c:243`.
type	string	mq_log

Example message

The following message is displayed on multiple lines, but IBM MQ typically writes the message on a single line.

```
{
  "ibm_messageId":"AMQ9209E",
  "ibm_arithInsert1":0,
  "ibm_arithInsert2":0,
  "ibm_commentInsert1":"localhost (127.0.0.1)",
  "ibm_commentInsert2":"TCP/IP",
  "ibm_commentInsert3":"SYSTEM.DEF.SVRCONN",
  "ibm_datetime":"2018-02-22T06:54:53.942Z",
  "ibm_serverName":"QM1",
  "type":"mq_log",
  "host":"0df0ce19c711",
  "loglevel":"ERROR",
  "module":"amqccita.c:4214",
  "ibm_sequence":"1519282493_947814358",
  "ibm_remoteHost":"127.0.0.1",
  "ibm_qmgrId":"QM1_2018-02-13_10.49.57",
  "ibm_processId":4927,
  "ibm_threadId":4,
  "ibm_version":"9.0.5.0",
  "ibm_processName":"amqrmppa",
  "ibm_userName":"johndoe",
  "ibm_installationName":"Installation1",
  "ibm_installationDir":"/opt/mqm",
  "message":"AMQ9209E: Connection to host 'localhost (127.0.0.1)' for channel
'SYSTEM.DEF.SVRCONN' closed."
}
```

MFT messages and return codes

Children of this topic are "Return codes for MFT" followed by diagnostic messages in numerical order, with Long Term Support (LTS) messages first followed by Continuous Delivery (CD) messages. The messages are further grouped according to the part of Managed File Transfer from which they originate.

For details of these messages, see IBM Documentation: https://www.ibm.com/docs/SSFKSJ_9.4.0/reference/mft_messages.html.

REST API messages

Reference information to help you identify and interpret the messages for the IBM MQ REST API. The messages are listed in numerical order, grouped according to the part of the API from which they originate.

For details of these messages, see IBM Documentation:

Messages for Long Term Support (LTS) releases	Messages for Continuous Delivery (CD) releases
• MQWB00xx: REST API messages • MQWB01xx: REST API messages • MQWB02xx: REST API messages • MQWB03xx: REST API messages • MQWB04xx: REST API messages • MQWB09xx: REST API messages • MQWB10xx: REST API messages • MQWB11xx: REST API messages	• MQWB00xx: REST API messages • MQWB01xx: REST API messages • MQWB02xx: REST API messages • MQWB03xx: REST API messages • MQWB04xx: REST API messages • MQWB09xx: REST API messages • MQWB10xx: REST API messages • MQWB11xx: REST API messages

Reading a message

The following information is provided for each message:

Message identifier
The message identifier is in three parts:

1. The characters "MQWB", which identify the message as being from the REST API
2. A four-digit decimal code
3. A suffix letter, indicating the severity of the message (I, W, E, S or T). See the following severity list.

Message text
A summary of the message

Severity

0: Informational
10: Warning
20: Error
30: Error
40: Severe
50: Terminating

Explanation
An explanation of the message giving further information.

Response
The response required from the user. In some cases, particularly for information messages, this might be "No action is required".

Message variables

Some messages display text or numbers that vary according to the circumstances giving rise to the message; these are known as *message variables*. The message variables are indicated as <insert_1>, <insert_2>, and so on.

In some cases a message might have variables in the Explanation or Response. Find the values of the message variables by looking in the error log. The complete message, including the Explanation and the Response, is recorded there.

ALW Telemetry messages

Reference information to help you identify and interpret the messages for IBM MQ Telemetry.

Messages for Long Term Support (LTS) releases	Messages for Continuous Delivery (CD) releases
• AMQP channel (AMQAM) messages (in IBM Documentation) • Telemetry (AMQCO) messages (in IBM Documentation) • Telemetry (AMQHT) messages (in IBM Documentation) • Telemetry (AMQXR) messages (in IBM Documentation) • Telemetry command utility (AMQXR) messages (in IBM Documentation)	• AMQP channel (AMQAM) messages (in IBM Documentation) • Telemetry (AMQCO) messages (in IBM Documentation) • Telemetry (AMQHT) messages (in IBM Documentation) • Telemetry (AMQXR) messages (in IBM Documentation) • Telemetry command utility (AMQXR) messages (in IBM Documentation)

Reading a message

The following information is provided for each message:

Message identifier
The message identifier is in three parts:

1. Five letters: "AMQAM", "AMQCO", "AMQHT", or "AMQXR", which identify the message as being from IBM MQ Telemetry.
2. A four-digit decimal code
3. A suffix letter, indicating the severity of the message (I, W, E, S or T). See the following severity list.

Message text
A summary of the message

Severity

0: Informational
10: Warning
20: Error
30: Error
40: Severe
50: Terminating

Explanation
An explanation of the message giving further information.

Response
The response required from the user. In some cases, particularly for information messages, this might be "No action is required".

Message variables

Some messages display text or numbers that vary according to the circumstances giving rise to the message; these are known as *message variables*. The message variables are indicated as <insert_1>, <insert_2>, and so on.

In some cases a message might have variables in the Explanation or Response. Find the values of the message variables by looking in the error log. The complete message, including the Explanation and the Response, is recorded there.

z/OS IBM MQ for z/OS messages, completion, and reason codes

Use this topic to interpret and understand the messages and codes issued by IBM MQ for z/OS.

The information in this topic can be used to understand a message or code produced by the IBM MQ for z/OS product. The topic is divided into the following parts:

"Messages for IBM MQ for z/OS" on page 291
Describes all IBM MQ messages in alphanumeric order.

All IBM MQ message identifiers are eight characters long. The first three characters are always CSQ. If you get a message with a different prefix, find out which product issued the message. See "Messages from other products" on page 1150.

The fourth character is the component identifier; this identifies the component of IBM MQ that issued the message. These are shown in "IBM MQ component identifiers" on page 1131. The fifth through seventh characters represent the numeric identifier, which is unique within the component. The last character is the message type code; this indicates the type of response that the message requires. Table 11 on page 289 shows the four type codes used by IBM MQ for z/OS.

Table 11. Message type codes

Type code	Response type	Response required
A	Immediate action	System operator action is required immediately. The associated task does not continue until the requested action has been taken.
D	Immediate decision	System operator decision or action is required immediately. The operator is requested to select from specific options, such as **retry** or **cancel**. The associated task does not continue until the requested decision has been made or action has been taken.
E	Eventual action	System operator action *will* be required; however, the associated task continues independently of system operator action.
I	Information only	No operator action is required. However, certain messages may be significant - please review Console message monitoring for further information.

In messages issued by the queue manager itself and the mover, the message identifier is normally followed by the *command prefix* (CPF); this indicates which IBM MQ queue manager issued the message. These messages have prefixes starting CSQE, CSQH, CSQI, CSQM, CSQN, CSQP, CSQR, CSQV, CSQX, CSQY, CSQ2, CSQ3, CSQ5, and CSQ9; some messages with prefixes CSQJ and CSQW also have the CPF. In certain exceptional cases, the CPF might show as blank.

Messages from CICS-related components (CSQC) show the CICS application ID or transaction ID if applicable.

Messages from other components, that is messages with prefixes CSQO, CSQQ, CSQU, and CSQ1 (and some with prefixes CSQJ and CSQW) have no indicator.

“IBM MQ for z/OS codes” on page 943
Describes all IBM MQ abend reason codes, and subsystem termination reason codes, in alphanumeric order.

The codes are four bytes long. The first byte is always 00; this is the high-order byte. The second byte is the hexadecimal identifier (Hex ID) of the IBM MQ component. These are shown in “IBM MQ component identifiers” on page 1131. The last two bytes are the numeric identifier, which is unique within the component.

“IBM MQ CICS adapter abend codes” on page 1131 and “IBM MQ CICS bridge abend codes” on page 1131
Describe the CICS abend codes issued by the IBM MQ CICS adapter, and the IBM MQ CICS bridge.

Accompanying each message and code is the following information, when applicable:

Explanation:
This section tells what the message or code means, why it occurred, and what caused it.

Severity:
Severity values have the following meanings:

0: An information message. No error has occurred.

4: A warning message. A condition has been detected of which the user should be aware. The user might need to take further action.

8: An error message. An error has been detected and processing could not continue.

12: A severe error message. A severe error has been detected and processing could not continue.

System action:
This part tells what is happening as a result of the condition causing the message or code. If this information is not shown, no system action is taken.

User response:
If a response by the user is necessary, this section tells what the appropriate responses are, and what their effect is. If this information is not shown, no user response is required.

Operator response:
If an operator response is necessary, this section tells what the appropriate responses are, and what their effect is. If this information is not shown, no operator response is required.

System programmer response:
If a response by the system programmer is required, this part tells what the appropriate responses are, and what their effect is. If this information is not shown, no system programmer response is required.

Programmer response:
If a programmer response is necessary, this part tells what the appropriate responses are, and what their effect is. If this information is not shown, no programmer response is required.

Problem determination:
This section lists the actions that can be performed to obtain adequate data for support personnel to diagnose the cause of the error. If this information is not shown, no problem determination is required.

Related reference
“AMQ messages on Multiplatforms” on page 244
The AMQ series of diagnostic messages are listed in this section in numerical order, grouped according to the part of IBM MQ from which they originate.

“Communications protocol return codes for z/OS” on page 1132
The communication protocols used by IBM MQ for z/OS can issue their own return codes. Use these tables to identify the return codes used by each protocol.

“Distributed queuing message codes” on page 1147

Distributed queuing is one of the components of IBM MQ for z/OS. Use this topic to interpret the message codes issued by the distributed queuing component.

“API completion and reason codes” on page 1151
For each call, a completion code and a reason code are returned by the queue manager or by an exit routine, to indicate the success or failure of the call.

“PCF reason codes” on page 1360
Reason codes might be returned by a broker in response to a command message in PCF format, depending on the parameters used in that message.

“Transport Layer Security (TLS) return codes” on page 1441
IBM MQ can use TLS with the various communication protocols. Use this topic to identify the error codes that can be returned by TLS.

“Transport Layer Security (TLS) return codes for z/OS” on page 1143
IBM MQ for z/OS can use TLS with the various communication protocols. Use this topic to identify the error codes that can be returned by TLS.

“WCF custom channel exceptions” on page 1446
Diagnostic messages are listed in this topic in numeric order, grouped according to the part of the Microsoft Windows Communication Foundation (WCF) custom channel from which they originate.

z/OS Messages for IBM MQ for z/OS

Each component of IBM MQ for z/OS can issue messages and each component uses a unique four character prefix for its messages. Use this topic to identify and interpret the messages for IBM MQ for z/OS components.

The following message types are described:

z/OS *Batch adapter messages (CSQB...)*

CSQB001E

Language environment programs running in z/OS batch or z/OS UNIX System Services must use the DLL interface to IBM MQ

Severity
4

Explanation

Application programs using IBM MQ and Language Environment® services from z/OS Batch or z/OS UNIX System Services must use the DLL interface to IBM MQ. This message is issued once per connection. The program which caused this message to be issued is using the stub interface to IBM MQ.

System action

Processing continues. The Async Consume feature of IBM MQ is not supported when using the non-DLL stub interface to IBM MQ.

z/OS *CICS adapter and Bridge messages (CSQC...)*

All the CICS versions supported by IBM MQ 9.0.0, and later, use the CICS supplied version of the bridge. See the DFHMQnnnn messages section of the CICS documentation for these messages.

z/OS *Coupling Facility manager messages (CSQE...)*

The value shown for *struc-name* in the coupling facility manager messages that follow is the 12-character name as used by IBM MQ. The external name of such CF structures for use by z/OS is formed by prefixing the IBM MQ name with the name of the queue sharing group to which the queue manager is connected.

CSQE005I
Structure *struc-name* connected as *conn-name*, version=*version*

Explanation

The queue manager has successfully connected to structure *struc-name*.

System action

Processing continues. The queue manager can now access the CF structure.

CSQE006I
Structure *struc-name* connection name *conn-name* disconnected

Explanation

The queue manager has disconnected from CF structure *struc-name*.

System action

Processing continues.

CSQE007I
event-type event received for structure *struc-name* connection name *conn-name*

Explanation

The queue manager has received XES event *event-type* for CF structure *struc-name*.

System action

Processing continues.

System programmer response

Examine the event code to determine what event was issued. The event codes are described in the *z/OS MVS Programming: Sysplex Services Reference* manual.

CSQE008I
Recovery event from *qmgr-name* received for structure *struc-name*

Explanation

The queue manager issued a peer level recovery event for CF structure *struc-name*.

System action

Processing continues. The queue manager will begin peer level recovery processing.

CSQE011I
Recovery phase 1 started for structure *struc-name* connection name *conn-name*

Explanation

Peer level recovery has started phase one of its processing, following the failure of another queue manager in the queue sharing group.

System action

Processing continues.

System programmer response

Determine why a queue manager within the queue sharing group failed.

CSQE012I
Recovery phase 2 started for structure *struc-name* connection name *conn-name*

Explanation

Peer level recovery has started phase two of its processing.

System action

Processing continues.

CSQE013I
Recovery phase 1 completed for structure *struc-name* connection name *conn-name*

Explanation

Peer level recovery has completed phase one of its processing.

System action

Processing continues.

CSQE014I
Recovery phase 2 completed for structure *struc-name* connection name *conn-name*

Explanation

Peer level recovery has completed phase two of its processing.

System action

Processing continues.

CSQE015I
Recovery phase 2 not attempted for structure *struc-name* connection name *conn-name*

Explanation

Phase two of peer level recovery processing was not attempted because of a previous error in phase one on one of the participating queue managers.

System action

Processing continues. The connection will be recovered by the failed queue manager when it restarts.

System programmer response

Investigate the cause of the error, as reported in the preceding messages.

CSQE016E
Structure *struc-name* connection name *conn-name* disconnected, RC=*return-code* reason=reason

Explanation

The queue manager has disconnected from CF structure *struc-name*.

System action

Processing continues.

System programmer response

Examine the return and reason codes to determine why the CF structure was disconnected. The codes are described in the *z/OS MVS Programming: Sysplex Services Reference* manual.

CSQE018I
Admin structure data building started

Explanation

The queue manager is building its own data for the administration structure.

System action

Processing continues.

CSQE019I
Admin structure data building completed

Explanation

The queue manager has built its own data for the administration structure.

System action

Processing continues.

CSQE020E
Structure *struc-name* connection as *conn-name* failed, RC=*return-code* reason= *reason* codes=*s1 s2 s3*

Explanation

The queue manager failed to connect to CF structure *struc-name*.

System action

This depends on the component that caused the connection request (queue manager or channel initiator) and the reason for connecting to the CF structure. The component might terminate, or might continue processing but with functions that require the structure inhibited.

System programmer response

Examine the return and reason codes to determine why the connect failed. Codes *s1 s2 s3* are the XES IXLCONN diagnosis codes, which are described in the *z/OS MVS Programming: Sysplex Services Reference* manual.

CSQE021I
Structure *struc-name* connection as *conn-name* warning, RC=*return-code* reason=*reason* codes=*s1 s2 s3*

Explanation

The queue manager has successfully connected to CF structure *struc-name*, but the XES IXLCONN call returned with a warning.

System action

Processing continues.

System programmer response

Examine the return and reason codes to determine why the connect warning message was issued. Codes *s1 s2 s3* are the XES IXLCONN diagnosis codes, which are described in the *z/OS MVS Programming: Sysplex Services Reference* manual.

CSQE022E
Structure *struc-name* unusable, size is too small

Explanation

The queue manager cannot use the named (coupling facility) (CF) structure because its size is less than the minimum that IBM MQ requires.

System action

The queue manager disconnects from the coupling facility (CF) structure, which becomes unusable. If it is an application structure, the queues that use the structure are not usable. If it is the administration structure, the queue manager terminates with completion code X'6C6' and reason code X'00C53000'.

System programmer response

Increase the size of the CF structure to at least the minimum size required. See Planning your coupling facility and offload storage environment for guidance on required structure sizes.

If the structure is allocated and the coupling facility Resource Manager policy allows the size of it to be increased, use the z/OS command SETXCF START,ALTER,STRNAME=*ext-struc-name*,SIZE=*newsize*. If the policy does not so allow, or there is insufficient space in the coupling facility that hosts the structure, the policy must be altered; then the structure can be rebuilt using the z/OS command SETXCF START,REBUILD,STRNAME=*ext-struc-name*. (In these commands, *ext-struc-name* is formed by prefixing *struc-name* with the queue sharing group name.)

If the structure is not allocated, alter the policy to specify a larger INITSIZE for the structure.

CSQE024E
Incorrect coupling facility level *level1*, required *level2*

Explanation

The queue manager cannot join the queue sharing group because the version of z/OS being used supports only CF level *level1*, but IBM MQ requires at least level *level2*.

System action

CF support is not active.

System programmer response

Upgrade z/OS and the coupling facility as necessary.

CSQE025E
Invalid UOW for *qmgr-name* in list *list-id* cannot be recovered, key=*uow-key*

Explanation

A unit-of-work descriptor was read during recovery processing that contained unexpected data. The descriptor was for the indicated queue manager; it was in the coupling facility list *list-id* and had key *uow-key* (shown in hexadecimal).

System action

The unit-of-work in error cannot be processed and the descriptor is marked as being in error. Processing continues.

System programmer response

Take a memory dump of the indicated list in your coupling facility administration structure for queue manager *qmgr-name* and contact IBM Support.

CSQE026E
Structure *struc-name* unusable, incorrect coupling facility level *level1*, required *level2*

Explanation

The queue manager cannot use the named CF structure because it has been allocated in a CF which supports level *level1*, but MQ requires at least level *level2*.

System action

The queues that use the CF structure are not usable.

System programmer response

Either upgrade the coupling facility, or use a CF structure which is in a CF running level *level2* or above.

CSQE027E
Structure *struc-name* unusable, vector size *n1* incorrect, required *n2*

Explanation

The queue manager cannot use the named CF structure because it has been allocated a list notification vector of size *n1*, but IBM requires at least size *n2*. This is probably because there is not enough available hardware storage area (HSA) for the vector.

System action

The queues that use the CF structure are not usable.

System programmer response

You cannot adjust the amount of HSA defined for your processor. Instead, retry the application (or other process) which was attempting to open the shared queue. If the problem persists, contact IBM Support for assistance.

CSQE028I
Structure *struc-name* reset, all messages discarded

Explanation

When it tried to connect to the named CF structure, the queue manager detected that the structure had been deleted, so a new empty structure has been created.

System action

All the messages on the queues that use the CF structure are deleted.

CSQE029E
Structure *struc-name* unusable, version *v1* differs from group version *v2*

Explanation

The queue manager cannot use the named CF structure because the version number of the structure differs from that of the queue sharing group.

System action

The queue manager disconnects from the CF structure, which becomes unusable. If it is an application structure, the queues that use the structure are not usable. If it is the administration structure, the queue manager terminates with completion code X'6C6' and reason code X'00C51057'.

System programmer response

Check that the configuration of your queue manager, queue sharing group, and data-sharing group is correct. If so, deallocate the CF structure using the z/OS commands **SETXCF FORCE,CON** and **SETXCF FORCE,STRUCTURE**. When you use these commands, the structure name is formed by prefixing *struc-name* with the queue sharing group name.

You might need to stop and restart the queue manager(s).

Note:

You can also use the **D XCF** command, for example **D XCF,STR,STRNAME=MQ7@CSQ_ADMIN** to show information about the structure and any connections.

CSQE030I
Serialized application cannot start, admin structure data incomplete

Explanation

A serialized application attempted to start, but it could not do so because one or more queue managers in the queue sharing group has not completed building its data for the administration structure. Messages CSQE031I and CSQE032I precede this message to identify such queue managers.

System action

The application is not started. The MQCONNX call that it issued to connect to the queue manager fails with a completion code of MQCC_FAILED and a reason code of MQRC_CONN_TAG_NOT_USABLE. (See “API completion and reason codes” on page 1151 for more information about these codes.)

System programmer response

The administration structure is automatically rebuilt. The rebuild can occur on any member of the QSG. Restart the application after the administration structure is successfully rebuilt, which is shown by message CSQE037I on the system performing the rebuild.

CSQE031I
Admin structure data from *qmgr-name* incomplete

Explanation

Some functions are not yet available because the indicated queue manager has not completed building its data for the administration structure.

System action

Processing continues. The functions will be available when all the queue managers identified by messages CSQE031I and CSQE032I have issued message CSQE019I.

CSQE032I
Admin structure data from *qmgr-name* unavailable

Explanation

Some functions are not yet available because the indicated queue manager is not active and therefore its data for the administration structure is not available.

System action

Processing continues.

System programmer response

The rebuild of the administration structure can occur on any member of the QSG. The functions will be available after the administration structures have been successfully rebuilt. Check the log for the messages CSQE036I and CSQE037I, which will indicate the start and completion of the administration structure rebuild.

CSQE033E
Recovery phase 1 failed for structure *struc-name* connection name *conn-name*, RC=*return-code* reason= *reason*

Explanation

An error occurred during phase one of peer level recovery processing. The recovery attempt is terminated. *return-code* and *reason* are the diagnosis codes (in hexadecimal) from an XES IXL call.

System action

Processing continues. The connection will be recovered by the failed queue manager when it restarts.

System programmer response

See the *z/OS MVS Programming: Sysplex Services Reference* manual for information about the XES IXL diagnosis codes. Restart the queue manager that failed; if it is unable to recover, contact IBM Support.

CSQE034E
Recovery phase 2 failed for structure *struc-name* connection name *conn-name*, RC=*return-code* reason=*reason*

Explanation

An error occurred during phase two of peer level recovery processing. The recovery attempt is terminated. *return-code* and *reason* are the diagnosis codes (in hexadecimal) from an XES IXL call.

System action

Processing continues. The connection will be recovered by the failed queue manager when it restarts.

System programmer response

See the *z/OS MVS Programming: Sysplex Services Reference* manual for information about the XES IXL diagnosis codes. Restart the queue manager that failed; if it is unable to recover, contact IBM Support.

CSQE035E
csect-name Structure *struc-name* in failed state, recovery needed

Explanation

The queue manager attempted to use CF structure *struc-name*, but it is in a failed state. The failure occurred previously; it was not caused by the current use of the structure.

System action

Processing continues, but queues that use this CF structure will not be accessible.

System programmer response

Check the console for messages from XES relating to the earlier failure, and investigate the cause. See the *z/OS MVS Programming: Sysplex Services Reference* manual for information about diagnosing problems in XES.

When the problem is resolved, issue a RECOVER CFSTRUCT command specifying TYPE(NORMAL) for this and any other failed CF structure.

CSQE036I
Admin structure data building started for *qmgr-name*

Explanation

The queue manager is building the indicated queue manager's data for the administration structure.

System action

Processing continues.

CSQE037I
Admin structure data building completed for *qmgr-name*

Explanation

The queue manager has built the indicated queue manager's data for the administration structure.

System action

Processing continues.

CSQE038E
Admin structure is full

Explanation

The queue manager cannot write to the administration structure in the coupling facility (CF) because it is full.

System action

The queue manager periodically retries the write attempt. If after a number of retries the structure is still full, this message is reissued and the queue manager terminates with a completion code X'5C6' and a reason code 00C53002.

System programmer response

Increase the size of the CF structure to at least the minimum size required. See the Defining coupling facility resources for guidance on required structure sizes.

If the structure is allocated and the coupling facility Resource Manager policy allows the size of it to be increased, use the z/OS command SETXCF START,ALTER,STRNAME=*ext-struc-name*,SIZE=*newsize*. If the policy does not allow this change, or there is insufficient space in the coupling facility that hosts the structure, the policy must be altered, then the structure can be rebuilt using the z/OS command SETXCF START,REBUILD,STRNAME=*ext-struc-name*. (In these commands, *ext-struc-name* is formed by prefixing CSQ_ADMIN with the queue sharing group name.)

If the structure is not allocated, alter the policy to specify a larger INITSIZE for the structure.

CSQE040I
Structure *struc-name* should be backed up

Explanation

The latest backup for the named CF structure is more than two hours old. Unless backups are taken frequently, the time to recover persistent messages on shared queues may become excessive.

The message is issued at checkpoint time if the queue manager was the one that took the last backup, or if it has used the structure since the last backup was taken.

System action

Processing continues.

System programmer response

Use the BACKUP CFSTRUCT command (on any queue manager in the queue sharing group) to make a new CF structure backup. You are recommended to set up a procedure to take frequent backups automatically.

CSQE041E
Structure *struc-name* backup is more than a day old

Explanation

The latest backup for the named CF structure is more than one day old. Unless backups are taken frequently, the time to recover persistent messages on shared queues might become excessive.

The message is issued at checkpoint time if the queue manager was the one that took the last backup, or if it has used the structure since the last backup was taken.

System action

Processing continues.

System programmer response

Use the BACKUP CFSTRUCT command (on any queue manager in the queue sharing group) to make a new CF structure backup. It is suggested you set up a procedure to take frequent backups automatically.

CSQE042E
csect-name Structure *struc-name* unusable, no EMC storage available

Explanation

The queue manager cannot use the named CF structure because its size is less than the minimum that IBM MQ requires. Specifically, the coupling facility allocation algorithms were unable to make any event monitor control (EMC) storage available during the allocation.

System action

The queue manager disconnects from the CF structure, and the CF structure becomes unusable. If it is an application structure, the queues that use the structure are not usable. If it is the administration structure, the queue manager terminates with completion code X'6C6' and reason code X'00C53003'.

System programmer response

Disconnect all connectors from the structure, and then issue

```
SETXCF FORCE,STR,STRNAME=name
```

to get the structure deallocated from the CF before you resize the structure.

Increase the size of the CF structure to at least the minimum size required. See Planning your coupling facility and offload storage environment for further information.

If the structure is allocated and the Coupling Facility Resource Manager policy allows the size of it to be increased, use the z/OS system command:

```
SETXCF START,ALTER,STRNAME=ext-struc-name,SIZE=newsize
```

If the CFRM policy does not allow an increase in size, or there is insufficient space in the coupling facility that hosts the structure, the policy must be altered. The structure can then be rebuilt using the z/OS system command:

```
SETXCF START,REBUILD,STRNAME=ext-struc-name
```

In these commands, *ext-struc-name* is formed by prefixing *struc-name* with the queue sharing group name.

If the structure is not allocated, alter the CFRM policy to specify a larger INITSIZE for the structure.

CSQE101I
csect-name Unable to back up or recover structure *struc-name*, structure in use

Explanation

A BACKUP or RECOVER CFSTRUCT command was issued, or automatic recovery started, for a CF structure that is in use by another process. The most likely cause is that another BACKUP or RECOVER CFSTRUCT command, or automatic recovery, is already in progress on one of the active queue managers in the queue sharing group.

This message can also be issued when new connections to the CF structure are being prevented by the system.

System action

Processing of the command, or automatic recovery for the identified structure, is terminated.

System programmer response

Check that the correct CF structure name was entered on the command. If so, wait until the current process ends before reissuing the command if required.

If there is no other BACKUP or RECOVER CFSTRUCT already in progress, check for previous messages that indicate why connections to the CF structure are being prevented.

CSQE102E
csect-name Unable to recover structure *struc-name*, not in failed state

Explanation

A RECOVER CFSTRUCT command was issued for a CF structure that is not in a failed state. Only a CF structure that has previously failed can be recovered.

System action

Processing of the command is terminated.

System programmer response

Check that the correct CF structure name was entered on the command.

CSQE103E
csect-name Unable to recover structures, admin structure data incomplete

Explanation

A RECOVER CFSTRUCT command was issued, but recovery could not be performed because one or more queue managers in the queue sharing group has not completed building its data for the administration structure.

System action

Messages CSQE031I and CSQE032I are sent to the z/OS console to identify such queue managers. Processing of the command is terminated.

System programmer response

The administration structure is automatically rebuilt. The rebuild can occur on any member of the QSG. Reissue the command after the administration structure is successfully rebuilt, which is shown by message CSQE037I on the system performing the rebuild.

CSQE104I
csect-name RECOVER task initiated for structure *struc-name*

Explanation

The queue manager has successfully started a task to process the RECOVER CFSTRUCT command for the named CF structure.

System action

Processing continues.

CSQE105I
csect-name BACKUP task initiated for structure *struc-name*

Explanation

The queue manager has successfully started a task to process the BACKUP CFSTRUCT command for the named CF structure.

System action

Processing continues.

CSQE106E
csect-name Unable to back up structure *struc-name*, reason=*reason*

Explanation

A BACKUP CFSTRUCT command was issued for a CF structure, but the backup could not be performed.

System action

Processing of the command is terminated.

System programmer response

Examine the reason code to determine why the CF structure could not be backed-up. The codes are described in "IBM MQ for z/OS codes" on page 943 and the *z/OS MVS Programming: Sysplex Services Reference* manual.

CSQE107E
csect-name Unable to back up or recover structure *struc-name*, structure has never been used

Explanation

A BACKUP or RECOVER CFSTRUCT command was issued, or automatic recovery started, for a CF structure that has never been used, and so does not contain any messages or data.

System action

Processing of the command, or automatic recovery for the identified structure, is terminated.

System programmer response

Check that the correct CF structure name was entered on the command.

CSQE108E
csect-name Unable to back up or recover structure *struc-name*, structure does not support recovery

Explanation

A BACKUP or RECOVER CFSTRUCT command was issued, or automatic recovery started, for a CF structure with a functional capability that is incompatible with this process; for example, the CF structure level is not high enough to support recovery, or the RECOVER attribute is set to NO.

System action

Processing of the command, or automatic recovery for the identified structure, is terminated.

System programmer response

Ensure that the CF structure is at a level of functional capability that allows the use of the BACKUP or RECOVER CFSTRUCT command and that its MQ RECOVER attribute is set to YES. You can check

the values using the DIS CFSTRUCT(*) ALL command. Check that the correct CF structure name was entered on the command.

CSQE109E
csect-name Unable to recover structure *struc-name*, no backup information available

Explanation

A RECOVER CFSTRUCT command was issued or automatic recovery started for a CF structure, but no backup information could be found.

System action

Processing of the command, or automatic recovery for the identified structure, is terminated.

System programmer response

Check that the correct CF structure name was entered on the command. If so, issue a BACKUP CFSTRUCT command to ensure that backup information is available.

CSQE110E
csect-name PURGE not allowed for structure *struc-name*

Explanation

A RECOVER CFSTRUCT command was issued for CF structure *struc-name* using TYPE(PURGE). This CF structure is a system application structure. To prevent loss of messages on system queues TYPE(PURGE) is not allowed for system application structures.

System action

Processing of the command is terminated.

System programmer response

Reissue the command without the TYPE(PURGE) option.

If structure recovery fails contact IBM Support.

CSQE111I

csect-name Structure *struct-name* will be set to failed state to allow recovery of failed SMDS data sets

Severity
0

Explanation

The **RECOVER CFSTRUCT** command was issued for a structure which is not in the failed state, but at least one of the related SMDS data sets is currently marked as failed, requiring recovery. The structure will be put into the failed state to make it unavailable for normal use so recovery can proceed.

System action

The structure is marked as failed and recovery processing continues.

CSQE112E
csect-name Unable to recover structure *struct-name*, failed to read required logs.

Explanation

A RECOVER CFSTRUCT command or automatic structure recovery was unable to read the logs required to recover a structure.

System action

Processing of the command is terminated.

Automatic recovery of the structure will not be attempted.

System programmer response

Check that the logs containing the RBA range indicated in message CSQE130I are available, and reissue the command.

Check for any prior errors or abends reporting problems using the logs.

Issue RECOVER CFSTRUCT(*struct-name*) to retry structure recovery.

CSQE120I
Backup of structure *struc-name* started at RBA=*rba*

Explanation

The named CF structure is being backed-up in response to a BACKUP CFSTRUCT command. The backup begins at the indicated RBA.

System action

Processing continues.

CSQE121I
csect-name Backup of structure *struc-name* completed at RBA=*rba*, size *n* MB

Explanation

The named CF structure has been backed-up successfully. The backup ends at the indicated RBA, and *n* is its approximate size in megabytes.

System action

Processing continues.

CSQE130I
Recovery of structure *struc-name* started, using *qmgr-name* log range from RBA=*from-rba* to RBA=*to-rba*

Explanation

CF structure recovery is starting in response to a RECOVER CFSTRUCT command. It must read the log range shown to determine how to perform recovery. The logs are read backwards, from the latest failure time of the structures to be recovered to the earliest last successful backup time of those structures.

System action

Processing continues.

CSQE131I
csect-name Recovery of structure *struc-name* completed

Explanation

The named CF structure has been recovered successfully. The structure is available for use again.

CF structure recovery was started in response to a RECOVER CFSTRUCT command. The log range determined how to perform recovery. The logs are read backwards, from the latest failure time of the structures to be recovered to the earliest last successful backup time of those structures.

System action

Processing continues.

CSQE132I
Structure recovery started, using log range from LRSN=*from-lrsn* to LRSN=*to-lrsn*

Explanation

CF structure recovery is starting in response to a RECOVER CFSTRUCT command. It must read the log range shown to determine how to perform recovery. The logs are read backwards, from the latest failure time of the structures to be recovered to the earliest last successful backup time of those structures.

See Recovering a CF structure for more information.

System action

Processing continues.

CSQE133I
Structure recovery reading log backwards, LRSN= *lrsn*

Explanation

This is issued periodically during log reading by CF structure recovery to show progress. The log range that needs to be read is shown in the preceding CSQE132I message.

CF structure recovery is starting in response to a RECOVER CFSTRUCT command. It must read the log range shown to determine how to perform recovery. The logs are read backwards, from the latest failure time of the structures to be recovered to the earliest last successful backup time of those structures.

System action

Processing continues.

System programmer response

If this message is issued repeatedly with the same LRSN value, investigate the cause; for example, IBM MQ might be waiting for a tape with an archive log data set to be mounted.

CSQE134I
Structure recovery reading log completed

Explanation

CF structure recovery is started in response to a RECOVER CFSTRUCT command. It must read the log range shown to determine how to perform recovery. The logs are read backwards, from the latest failure time of the structures to be recovered, to the earliest last successful backup time of those structures.

CF structure recovery has completed reading the logs. The individual structures can now be recovered.

System action

Each CF structure is recovered independently, as shown by messages CSQE130I and CSQE131I.

CSQE135I
Recovery of structure *struc-name* reading log, RBA=*rba*

Explanation

This is issued periodically during log reading for recovering the named CF structure to show progress. The log range that needs to be read is shown in the preceding CSQE130I message.

System action

Processing continues.

System programmer response

If this message is issued repeatedly with the same RBA value, investigate the cause; for example, MQ might be waiting for a tape with an archive log data set to be mounted.

CSQE136I

Error returned by Db2 when clearing queue *queue-name*, list header number=*list header number*, structure number=*strucnum*

Severity
4

Explanation

Shared queue messages greater than 63 KB in size have their message data held as one or more binary large objects (BLOBs) in a Db2 table. An error was returned by Db2 when clearing these messages from the table.

Note that the list header number, and structure number, are output in hexadecimal format.

System action

Processing continues.

System programmer response

The messages have been deleted from the coupling facility but message data might remain in Db2 as orphaned BLOBs. This message is normally preceded by message CSQ5023E. Examine the Db2 job log to determine why the error occurred. The orphaned messages can be deleted by issuing the '**DISPLAY GROUP OBSMSGS(YES)**' command after 24 hours.

CSQE137E

csect-name Db2 and CF structure out of sync for queue *queue-name*, list header number=*list header number*, structure number=*strucnum*

Severity
4

Explanation

The queue manager has identified a discrepancy between the information stored about a queue in the coupling facility and the corresponding information in Db2.

Note that the list header number, and structure number, are output in hexadecimal format.

System action

Processing continues, but applications are unable to open the affected queue until the discrepancy is resolved by the System Programmer.

System programmer response

If the queue manager has recently been recovered from a backup then the recovery process should be reviewed to ensure that everything was correctly restored, including any Db2 tables associated with the queue manager.

If the cause of the problem cannot be determined then contact IBM Support for assistance.

CSQE138I
csect-name Structure struc-name is already in the failed state

Explanation

A **RESET CFSTRUCT ACTION(FAIL)** command was issued for a CF structure that is already in the failed state.

System action

Processing of the command is terminated.

CSQE139I
csect-name Unable to fail structure struc-name, structure in use

Explanation

A **RESET CFSTRUCT ACTION(FAIL)** command was issued for a CF structure that is in use by another process

System action

Processing of the command is terminated.

System programmer response

Check that the correct CF structure name was entered on the command. If so, wait until the process ends before reissuing the command if required.

CSQE140I

csect-name Started listening for ENF 35 events for structure *structure-name*

Severity
0

Explanation

The queue manager has registered to receive ENF 35 events and will attempt to reconnect to the identified structure if it is notified that a coupling facility resource has become available.

System action

Processing continues.

CSQE141I
csect-name Stopped listening for ENF 35 events for structure *structure-name*

Explanation

The queue manager has de-registered from receiving ENF 35 events for the identified structure, and will not attempt to reconnect to it if notified that a coupling facility resource has become available.

System action

Processing continues.

CSQE142I
csect-name Total loss of connectivity reported for structure *structure-name*

Explanation

The queue manager has been notified that no systems in the sysplex have connectivity to the coupling facility in which the identified structure is allocated.

System action

If automatic recovery has been enabled for the identified structure one of the queue managers in the queue sharing group will attempt to recover the structure in an alternative coupling facility, if one is available.

System programmer response

Investigate and resolve the loss of connectivity to the coupling facility on which the structure is allocated.

CSQE143I
csect-name Partial loss of connectivity reported for structure *structure-name*

Explanation

The queue manager has lost connectivity to the coupling facility in which the identified structure is allocated, and has been notified that the coupling facility is still available on other systems in the sysplex.

System action

A system-managed rebuild will be scheduled to rebuild the structure in an alternative coupling facility, if one is available.

System programmer response

Investigate and resolve the loss of connectivity to the coupling facility on which the structure is allocated.

CSQE144I
csect-name System-managed rebuild initiated for structure *structure-name*

Explanation

The queue manager has initiated a system-managed rebuild for the identified structure on an alternative coupling facility.

System action

Processing continues and when the process has completed, you receive message CSQE005I.

CSQE145E

csect-name Auto recovery for structure *structure-name* is not possible, no alternative CF defined in CFRM policy

Severity
8

Explanation

The queue manager has lost connectivity to the coupling facility in which the identified structure is allocated, but cannot automatically recover the structure because there is no alternative coupling facility in the CFRM preference list.

System action

Processing continues without connectivity to the structure. Any queues that reside on the application structure remain unavailable.

System programmer response

Investigate and resolve the loss of connectivity to the Coupling Facility on which the structure is allocated.

CSQE146E

csect-name System-managed rebuild for structure *structure-name* failed, reason=*reason*

Severity
8

Explanation

The queue manager attempted to initiate a system-managed rebuild for the identified structure but the rebuild could not be performed.

System action

Processing continues without connectivity to the structure. Any queues that reside on the application structure remain unavailable.

System programmer response

Examine the reason code to determine why the system-managed rebuild could not be completed. The codes are described in the *z/OS MVS Programming: Sysplex Services Reference* manual.

CSQE147I
csect-name System-managed rebuild for structure *structure-name* is already in progress

Explanation

The queue manager attempted to initiate a system-managed rebuild for the identified structure but determined that another queue manager in the queue sharing group has initiated it already.

System action

Processing continues.

CSQE148I
csect-name Loss of connectivity processing for structure *structure-name* deferred

Explanation

The queue manager has lost connectivity to the coupling facility in which the identified structure is allocated, but MVS has requested that the queue manager should not take action until a subsequent notification is received.

System action

Processing continues without connectivity to the structure. Any queues that reside on the application structure remain unavailable.

CSQE149I
csect-name Waiting for other queue managers to disconnect from structure *structure-name*

Explanation

The queue manager has lost connectivity to the coupling facility, in which the identified structure is allocated, but cannot delete the structure or initiate a system-managed rebuild because one or more queue managers that also lost connectivity remain connected to it.

System action

The queue manager will periodically retry the attempted operation until all of the queue managers have disconnected.

CSQE150I
csect-name System-managed rebuild already completed for structure *structure-name*

Explanation

A system-managed rebuild for the identified structure is unnecessary as another request to rebuild the structure has been completed.

System action

Processing continues.

CSQE151I
csect-name Loss of admin structure connectivity toleration enabled

Explanation

If any queue manager in the queue sharing group loses connectivity to the administration structure the structure will be rebuilt in an alternative CF, if one is available.

If the structure cannot be rebuilt, some shared queue functions on queue managers that have lost connectivity will be unavailable until connectivity to the structure has been restored. Access to private queues will not be affected.

System action

Processing continues.

CSQE152I
csect-name Loss of admin structure connectivity toleration disabled

Explanation

If the queue manager loses connectivity to the administration structure no attempt to rebuild it is made. The queue manager terminates with abend code 5C6-00C510AB.

This can occur if the CFCONLOS queue manager attribute is set to TERMINATE.

System action

Processing continues.

CSQE153I
csect-name Auto recovery for structure *struct-name* has been scheduled

Explanation

The queue manager has detected that the identified structure which has automatic recovery enabled, has failed, or connectivity to it has been lost on all systems in the sysplex.

The queue manager has scheduled an attempt to recover the structure.

System action

One of the active queue managers in the queue sharing group will attempt to recover the identified structure.

CSQE154I
csect-name Structure *struct-name* has been deleted

Explanation

The queue manager has successfully deleted the identified structure from the coupling facility.

System action

Processing continues.

CSQE155I
csect-name Structure *struct-name* has already been deleted

Explanation

The queue manager attempted to delete the identified structure from the coupling facility. It could not be deleted because it was not allocated.

System action

Processing continues.

CSQE156I
csect-name Structure *struct-name* has already been reallocated

Explanation

The queue manager lost connectivity to the identified structure. When attempting to delete the structure the queue manager found that the structure had been reallocated since connectivity was lost.

System action

Processing continues.

CSQE157E

csect-name Unable to recover structure *struc-name*, no suitable CF available

Severity
8

Explanation

A RECOVER CFSTRUCT command was issued or automatic recovery started for the identified structure, but there was no suitable Coupling Facility available in which to allocate it.

System action

Processing of the command, or automatic recovery for the identified structure, is terminated.

System programmer response

Ensure that a suitable Coupling Facility in the CFRM preference list for the identified structure is available, then reissue the command.

CSQE158E

csect-name Recovery of structure *struc-name* failed, reason=*reason*

Severity
8

Explanation

Recovery of the identified (coupling facility) CF structure has failed.

System action

Processing continues, but queues that use the identified (coupling facility) CF structure will not be accessible.

System programmer response

Refer to coupling facility codes (X'C5') for information about the reason code. Use this information to solve the problem, then reissue the RECOVER CFSTRUCT command for structures that do not have automatic recovery enabled.

CSQE159I

csect-name Waiting for structure rebuild to complete for structure *structure-name*

Explanation

The queue manager has lost connectivity to the coupling facility, in which the identified structure is allocated, but cannot delete the structure or initiate a system-managed rebuild, because a structure rebuild is currently in progress.

System action

The queue manager will periodically retry the attempted operation, until the structure rebuild is finished.

CSQE160I

csect-name Auto recovery for structure *struc-name* is suspended

Explanation

The queue manager detected that recovery for structure *struc-name* is not possible. Automatic recovery of the structure is suspended.

System action

Automatic recovery for structure *struc-name* is suspended. Automatic recovery is resumed when a successful connection to the structure is established.

System programmer response

Check for any previous errors or abends reporting problems recovering the structure.

Issue RECOVER CFSTRUCT(*struct-name*) to retry structure recovery.

CSQE161E

csect-name queue sharing group state is inconsistent; no XCF data for queue manager *qmgr-number*

Explanation

A RECOVER CFSTRUCT command or automatic structure recovery could not read all the log data required for recovery, because there was no XCF data for one of the queue managers in the QSG. *qmgr-number* is the number of the affected queue manager in the MQ Db2 tables.

System action

Processing of the command is terminated. Automatic recovery of the structure will not be attempted.

System programmer response

If the queue manager with number *qmgr-number* in the MQ Db2 tables has been force removed from the queue sharing group then added back into the QSG, start the queue manager and issue the

RECOVER CFSTRUCT command again. Otherwise, reset the structure to an empty state by issuing the RECOVER CFSTRUCT TYPE(PURGE) command.

CSQE162E

csect-name Structure *struc-name* could not be deleted, RC=*return-code* reason=*reason*

Severity

8

Explanation

The queue manager failed to delete structure *struc-name* from the Coupling Facility when processing a DELETE CFSTRUCT command.

System action

Processing continues.

System programmer response

Examine the return and reason codes to determine why the Coupling Facility structure could not be deleted by the IXLFORCE macro. The codes are described in the *z/OS MVS Programming: Sysplex Services Reference* manual.

Correct the problem that caused the failure, then delete the structure by issuing the SETXCF FORCE,STRUCTURE z/OS command.

CSQE201E

Media manager request failed with return code *ccccffss* processing *req* request for control interval *rci* in SMDS(*qmgr-name*) CFSTRUCT(*struc-name*) data set *dsname*

Severity

8

Explanation

An error occurred when attempting the indicated media manager request (READ, UPDATE or FORMAT) for the data set.

ccccffss

is the media manager return code in hexadecimal. The last byte *ss* indicates the overall type of error:

08

Extent error

0C

Logic error

10

Permanent I/O error

14

Undetermined error

The *cccc* field identifies the specific error and the *ff* field identifies the function which returned the error. See the *z/OS DFSMSdfp Diagnosis* manual for further details of media manager return codes.

req

specifies the type of request:

READ

Read one or more control intervals.

UPDATE

Rewrite one or more control intervals.

FORMAT

Format one or more control intervals.

rci
identifies the relative control interval (RCI) number of the control interval being accessed, in hexadecimal.

qmgr-name
identifies the queue manager which owns the shared message data set.

struc-name
identifies the application structure associated with the shared message data set.

dsname
shows the full name of the shared message data set.

System action

This typically results in the **SMDS** status being set to **FAILED** (if it is the data set owned by the current queue manager) or the **SMDSCONN** status being set to **ERROR** (if it is a data set owned by a different queue manager).

System programmer response

If the problem is a permanent I/O error caused by damage to the data set and recovery logging was enabled, the data set can be recovered by the recreating it from a backup and reapplying the logged changes using the **RECOVER CFSTRUCT** command.

If the data set is temporarily unavailable (for example because of a device connectivity problem) but is not damaged, then when the data set is available again, it can be put back into normal use by using the **RESET SMDS** command to set the status to **RECOVERED**.

CSQE202E

Media manager service failed with return code *ret-code*, feedback code *feedback-code*, processing *function* for SMDS(*qmgr-name*) CFSTRUCT(*struc-name*) data set *dsname*

Severity
8

Explanation

A media manager support services (MMGRSRV) function gave an unexpected error.

ret-code
indicates the MMGRSRV return code, in hexadecimal.

08
Media Manager Services error.

14
Indeterminate error

feedback-code
indicates the 8-byte MMGRSRV internal feedback code, in hexadecimal.

For CONNECT processing, the first byte of this feedback code is the same as the VSAM OPEN error information returned in ACBERFLG.

function
indicates the type of function requested, which can be any of the following:

CONNECT
Open the data set.

DISCONNECT
Close the data set.

EXTEND
Extend the data set being written by the current queue manager, or obtain access to recently added extents for a data set which has been extended by another queue manager.

CATREAD
Obtain the highest allocated and highest used control interval numbers from the catalog entry for the current data set.

CATUPDT
Update the highest used control interval in the catalog entry for the current data set, after formatting new extents.

qmgr-name
identifies the queue manager which owns the shared message data set.

struc-name
identifies the application structure associated with the shared message data set.

dsname
shows the full name of the shared message data set.

System action

This typically results in the **SMDS** status being set to **FAILED** (if it is the data set owned by the current queue manager) or the **SMDSCONN** status being set to **ERROR** (if it is a data set owned by a different queue manager).

System programmer response

This message is normally preceded by a system message such as IEC161I from VSAM or DFP indicating the nature of the error.

If the problem is a permanent I/O error caused by damage to the data set and recovery logging was enabled, the data set can be recovered by the recreating it from a backup and reapplying the logged changes using the **RECOVER CFSTRUCT** command.

If the data set is temporarily unavailable (for example because of a device connectivity problem) but is not damaged, then when the data set is available again, it can be put back into normal use by using the **RESET SMDS** command to set the status to **RECOVERED**.

CSQE211I

Formatting is in progress for *count* pages in SMDS(*qmgr-name*) CFSTRUCT(*struc-name*) data set *dsname*

Severity
0

Explanation

The data set is being formatted from the current highest used page to the highest allocated page. This message occurs either when a new extent has been allocated or immediately after opening an existing data set which has not been fully formatted (that is, the highest used page is less than the highest allocated page).

count
indicates the number of pages which need to be formatted (in decimal).

qmgr-name
identifies the queue manager which owns the shared message data set.

struc-name
identifies the application structure associated with the shared message data set.

dsname
shows the full name of the shared message data set.

System action

Formatting continues.

CSQE212I

Formatting is complete for SMDS(*qmgr-name*) CFSTRUCT(*struc-name*) data set *dsname*

Severity
0

Explanation

Formatting of the data set has completed and the highest used page has been successfully updated in the catalog.

dsname
identifies the shared message data set.

struc-name
identifies the application structure associated with the shared message data set.

System action

The newly formatted space is made available for use.

CSQE213I

SMDS(*qmgr-name*) CFSTRUCT(*struc-name*) data set *dsname* is now *percentage*% full

Severity
0

Explanation

The data set is nearly full.

qmgr-name
identifies the queue manager which owns the shared message data set.

struc-name
identifies the application structure associated with the shared message data set.

dsname
shows the full name of the shared message data set.

percentage
shows the percentage of data blocks in the data set which are currently in use.

This message is issued when the data set becomes 90% full, 92% full, and so on, up to 100%. After this message has been issued for a particular percentage, it is not issued again until the usage has changed in either direction by at least 2%. If the usage then decreases to 88% or less (as a result of messages being deleted or as a result of the data set being expanded) a final message is issued to indicate the new usage percentage.

System action

If expansion is allowed, the data set is expanded. If the data set reaches 100% full, then requests to put new messages that require space in the data set are rejected with return code MQRC_STORAGE_MEDIUM_FULL.

System programmer response

You can check the usage in more detail using the **DISPLAY USAGE** command with the **SMDS** keyword.

CSQE215I

Further expansion of SMDS(*qmgr-name*) CFSTRUCT(*struc-name*) data set *dsname* is not possible because the maximum number of extents have been allocated

Severity
0

Explanation

The media manager interface has indicated that the data set has reached the maximum number of extents, and cannot be expanded any further.

qmgr-name
identifies the queue manager that owns the shared message data set.

struc-name
identifies the application structure associated with the shared message data set.

dsname
shows the full name of the shared message data set.

This message can be issued when the data set is opened, or following an expansion attempt, which might have been successful, as indicated by previous messages.

System action

The expansion option for the data set is changed to **DSEXPAND(NO)** to prevent further expansion attempts.

System programmer response

The only way to expand the data set further is to make it temporarily unavailable by using the **RESET SMDS** command to mark the status as **FAILED**, copy it to a new location using larger extents, then make it available again using the **RESET SMDS** command to mark the status as **RECOVERED**.

CSQE217I

Expansion of SMDS(*qmgr-name*) CFSTRUCT(*struc-name*) data set *dsname* was successful, *count* pages added, total pages *total*

Severity
0

Explanation

The data set was expanded, and one or more new extents have been successfully added.

qmgr-name
identifies the queue manager, which owns the shared message data set.

struc-name
identifies the application structure associated with the shared message data set.

dsname
shows the full name of the shared message data set.

count
indicates the number of new pages that have been allocated (in decimal).

total
indicates the total number of pages currently allocated (in decimal).

System action

The queue manager formats the newly allocated space.

CSQE218E

Expansion of SMDS(*qmgr-name*) CFSTRUCT(*struc-name*) data set *dsname* was unsuccessful

Severity
8

Explanation

An attempt was made to expand the data set, but it was unsuccessful, typically because insufficient space was available.

qmgr-name
identifies the queue manager, which owns the shared message data set.

struc-name
identifies the application structure associated with the shared message data set.

dsname
shows the full name of the shared message data set.

System action

The expansion option for the data set is changed to **DSEXPAND(NO)** to prevent further expansion attempts.

System programmer response

Check for messages from VSAM or DFP that explain why the request was unsuccessful, and do the required actions.

If space is made available later, change the expansion option back to allow expansion to be tried again.

CSQE219I

Extents refreshed for SMDS(*qmgr-name*) CFSTRUCT(*struc-name*) data set *dsname*, *count* pages added, total pages *total*

Severity
0

Explanation

The data set was extended by another queue manager. The current queue manager used media manager services to update the extent information for the open data set to read message data within the new extents.

qmgr-name
identifies the queue manager that owns the shared message data set.

struc-name
identifies the application structure associated with the shared message data set.

dsname
shows the full name of the shared message data set.

count
indicates the number of new page that have been allocated (in decimal).

total
indicates the total number of pages currently allocated (in decimal).

System action

The new extents are made visible to the current queue manager.

CSQE222E

Dynamic allocation of SMDS(*qmgr-name*) CFSTRUCT(*struc-name*) data set *dsname* failed with return code *ret-code*, reason code *eeeeiiii*

Severity
8

Explanation

An attempt was made to allocate the data set using the data set name formed by taking the generic **DSGROUP** name and inserting the queue manager name, but the DYNALLOC macro returned an error.

qmgr-name
identifies the queue manager which owns the shared message data set.

struc-name
identifies the application structure associated with the shared message data set.

dsname
shows the full name of the shared message data set.

ret-code
shows the return code from DYNALLOC, in decimal.

eeeeiiii
shows the reason code, consisting of the error and information codes returned by DYNALLOC, in hexadecimal.

For more information on the reason codes that are populated to the CSQE222E message, see z/OS MVS Programming: Authorized Assembler Services Guide .

System action

This typically results in the **SMDS** status being set to **FAILED** (if it is the data set owned by the current queue manager) or the **SMDSCONN** status being set to **ERROR** (if it is a data set owned by a different queue manager).

System programmer response

Check the job log for dynamic allocation error messages giving more details about the problem.

After any changes, use the **START SMDSCONN** command to trigger a new attempt to use the data set.

When the reason code is '02540000', indicating that the allocation failed due to a required ENQ being unavailable, the queue manager will automatically retry the allocation request on subsequent attempts to access the SMDS.

When the reason code is '02380000', indicating that the allocation failed because there was not enough space in the MVS task I/O table (TIOT), increase the size of the TIOT using the ALLOCxx PARMLIB member.

Alternatively, from IBM MQ 9.4.0, specify NON_VSAM_XTIOT=YES in the DEVSUPxx PARMLIB member which allows use of the extended TIOT.

CSQE223E

Dynamic deallocation of SMDS(*qmgr-name*) CFSTRUCT(*struc-name*) data set *dsname* failed with return code *ret-code*, reason code *eeeeiiii*

Severity
8

Explanation

An attempt was made to deallocate the data set but the DYNALLOC macro returned an error.

qmgr-name
identifies the queue manager which owns the shared message data set.

struc-name
identifies the application structure associated with the shared message data set.

dsname
shows the full name of the shared message data set.

ret-code
shows the return code from DYNALLOC, in decimal.

eeeeiiii
shows the reason code, consisting of the error and information codes returned by DYNALLOC, in hexadecimal.

System action

No further action is taken, but problems can occur if an attempt is made to use the data set, either from another job or from the same queue manager.

System programmer response

Check the job log for dynamic allocation error messages giving more details about the problem.

CSQE230E

csect-name SMDS(*qmgr-name*) CFSTRUCT(*struc-name*) data set *dsname* saved space map cannot be used the time stamp *time1* does not match the last CLOSE time stamp *time2* in the SMDS object

Severity
8

Explanation

The shared message data set owned by this queue manager appears to have been closed normally last time it was used, with a saved space map, but the time stamp in the data set does not match the time stamp stored in the SMDS object in Db2 the last time this queue manager closed the data set. This means that the saved space map may not be consistent with the current messages in the coupling facility, so it needs to be rebuilt.

The most probable cause for this message is that the data set has been copied or restored from a copy which was not completely up to date.

qmgr-name
identifies the queue manager that owns the shared message data set.

struc-name
identifies the application structure associated with the shared message data set.

dsname
shows the full name of the shared message data set.

time1
shows the time stamp found in the data set header.

time2
shows the time stamp found in the SMDS object in Db2.

System action

The existing saved space map is ignored and the space map is rebuilt by scanning the messages in the coupling facility structure which refer to the data set.

The rebuild scan process keeps track of the most recent message in the coupling facility that refers to the data set, and at the end of the scan it checks that the matching message data is found in the data set. If so, it is assumed that all changes up to at least that time are present in the data set, so no data has been lost, and the data set can be opened normally. Otherwise, message CSQI034E is issued and the data set is marked as failed.

CSQE231E

SMDS(*qmgr-name*) CFSTRUCT(*struc-name*) data set *dsname* cannot be used because it is not a VSAM linear data set with control interval size 4096 and SHAREOPTIONS(2 3)

Severity
8

Explanation

The specified data set is not a VSAM linear data set, or the control interval size is not the default value 4096, or the wrong sharing options have been specified.

qmgr-name
identifies the queue manager that owns the shared message data set.

struc-name
identifies the application structure associated with the shared message data set.

dsname
shows the full name of the shared message data set.

If the data set was initially empty, the sharing options are not checked until the data set has been initialized, closed, and reopened.

System action

The data set is closed and the **SMDS** status is set to **FAILED**.

System programmer response

Delete the incorrect data set, and create a one of the same name with the correct attributes.

After any changes, use the **START SMDSCONN** command to trigger a new attempt to use the data set.

CSQE232E

csect-name SMDS(*qmgr-name*) CFSTRUCT(*struc-name*) data set *dsname* cannot be used because the identification information (*field-name*) in the header record is incorrect

Severity

8

Explanation

When the data set was opened, there was existing information in the header record (so the data set was not newly formatted) but the information did not match the expected data set identification. The identification information includes a marker "CSQESMDS" for a shared message data set followed by the names of the queue sharing group, the application structure and the queue manager which owns the shared message data set.

qmgr-name
identifies the queue manager that owns the shared message data set.

struc-name
identifies the application structure associated with the shared message data set.

dsname
shows the full name of the shared message data set.

field-name
identifies the first header identification field which did not have the expected value.

System action

The data set is closed and the connection is marked as **AVAIL(ERROR)**. If the data set status is **ACTIVE** or **RECOVERED**, indicating that it was currently in use, the status is changed to **FAILED**.

System programmer response

If the data set was already in use, this probably indicates that it has been overwritten in some way, in which case any persistent messages can be recovered using the **RECOVER CFSTRUCT** command.

If the data set was not yet in use, or was currently empty, ensure that it is either formatted or emptied before trying to use it again. After any changes, use the **START SMDSCONN** command to trigger a new attempt to use the data set.

To display the data set header record, you can use the Access Method Services **PRINT** command, for example as follows:

```
PRINT INDATASET('dsname') TOADDRESS(4095)
```

The format of the identification information within the data set header record is as follows:

Table 12. Format of identification information within the data set header record.

Offset: Dec	Offset: Hex	Type	Length	Field	Description
8	8	Character	8	MARKER	Marker 'CSQESMDS'
16	10	Character	4	Queue sharing group	Queue sharing group name
20	14	Character	12	CFSTRUCT	Structure name
3	20	Character	4	SMDS	Owning queue manager

Table 12. Format of identification information within the data set header record. (continued)					
Offset: Dec	**Offset: Hex**	**Type**	**Length**	**Field**	**Description**
36	24	Integer	4	VERSION	Header version 1

CSQE233E

SMDS(*qmgr-name*) CFSTRUCT(*struc-name*) data set *dsname* cannot be used because the header record indicates a newly formatted data set but it was already being used

Severity
8

Explanation

When the data set was opened, the identification information in the header record was zero, indicating a new empty data set, but the data set was already in use, so it should not now be empty.

qmgr-name
identifies the queue manager that owns the shared message data set.

struc-name
identifies the application structure associated with the shared message data set.

dsname
shows the full name of the shared message data set.

System action

The data set is closed and marked as **FAILED**.

System programmer response

Any persistent messages can be recovered using the **RECOVER CFSTRUCT** command.

CSQE234I

SMDS(*qmgr-name*) CFSTRUCT(*struc-name*) data set *dsname* was empty so it requires formatting

Severity
0

Explanation

When the data set was opened, it was found to be empty, with no existing data and no pre-formatted space. In this case, VSAM does not allow shared access to the data set. The queue manager needs to initialize the data set.

qmgr-name
identifies the queue manager that owns the shared message data set.

struc-name
identifies the application structure associated with the shared message data set.

dsname
shows the full name of the shared message data set.

System action

The data set is pre-formatted up to the end of the existing extents. There is a short delay before the data set is fully available.

CSQE235I

SMDS(*qmgr-name*) CFSTRUCT(*struc-name*) data set *dsname* was not fully formatted so it requires additional formatting

Severity
0

Explanation

This occurs if the existing data set extents have not been fully formatted when the data set is opened.

qmgr-name
identifies the queue manager that owns the shared message data set.

struc-name
identifies the application structure associated with the shared message data set.

dsname
shows the full name of the shared message data set.

System action

The data set is formatted up to the end of the existing extents. There is a short delay before the data set is fully available.

CSQE236I

SMDS(*qmgr-name*) CFSTRUCT(*struc-name*) data set *dsname* cannot be used because there is not enough main storage available to build the space map

Severity

8

Explanation

The queue manager needs to build a space map in main storage to manage the free space in the data set, but it was unable to obtain sufficient main storage.

qmgr-name
identifies the queue manager which owns the shared message data set.

struc-name
identifies the application structure associated with the shared message data set.

dsname
shows the full name of the shared message data set.

System action

The data set is not opened.

System programmer response

Consider increasing the queue manager's MEMLIMIT.

If necessary, use the START SMDSCONN command to request another attempt to open the data set.

For more details see Address space storage.

CSQE237I

SMDS(*qmgr-name*) CFSTRUCT(*struc-name*) data set *dsname* cannot be extended because there is not enough main storage available to build the space map

Severity
8

Explanation

The queue manager needs to build space map blocks in main storage to manage the additional space in the extended data set, but it was unable to obtain sufficient main storage.

qmgr-name
identifies the queue manager which owns the shared message data set.

struc-name
identifies the application structure associated with the shared message data set.

dsname
shows the full name of the shared message data set.

System action

The new extents of the data set are not available for use.

System programmer response

Consider increasing the queue manager's MEMLIMIT.

If necessary, use the START SMDSCONN command to request another attempt to open the data set.

For more details see Address space storage.

CSQE238I

SMDS(*qmgr-name*) CFSTRUCT(*struc-name*) data set *dsname* is too small to use because the initial space allocation is less than two logical blocks

Severity
8

Explanation

The minimum supported data set size requires at least one logical block for control information and one logical block for data, but the data set is smaller than two logical blocks.

qmgr-name
identifies the queue manager which owns the shared message data set.

struc-name
identifies the application structure associated with the shared message data set.

dsname
shows the full name of the shared message data set.

System action

The data set is not opened.

System programmer response

Delete the data set and re-create it with a larger space allocation.

After making changes, use the **START SMDSCONN** command to request another attempt to open the data set.

CSQE239I

SMDS(*qmgr-name*) CFSTRUCT(*struc-name*) data set *dsname* has become full so new large messages can no longer be stored in it

Severity
8

Explanation

A message written to a shared queue contains data which is large enough to require offloading to a data set, but there is insufficient space in the data set. Further requests are likely to fail until existing messages have been read and deleted from the data set.

qmgr-name
identifies the queue manager which owns the shared message data set.

struc-name
identifies the application structure associated with the shared message data set.

dsname
shows the full name of the shared message data set.

System action

Any requests encountering this problem are rejected with MQRC_STORAGE_MEDIUM_FULL. This message is not issued again until the data set has been below 90% full since the previous time it was issued.

System programmer response

This problem means that the backlog of unprocessed large shared messages exceeds the size of the data set, but the data set could not be extended in time to avoid the problem.

Ensure that applications to remove large messages from the shared queues are running. Check also for previous problems relating to extending the data set, for example if there was insufficient space on eligible volumes.

CSQE241I

SMDS(*qmgr-name*) CFSTRUCT(*struc-name*) now has STATUS(*status*)

Severity

0

Explanation

The status of the shared message data set for the specified queue manager and application structure has been changed to the indicated value, either by automatic status management or by a **RESET SMDS** command.

qmgr-name
identifies the queue manager that owns the shared message data set.

struc-name
identifies the application structure associated with the shared message data set.

status
shows the new status value. For details of specific status values, see the DISPLAY CFSTATUS command with the **TYPE(SMDS)** option.

System action

All queue managers connected to the structure are notified of the status change. The queue managers take appropriate action if necessary, for example opening or closing the data set.

CSQE242I

SMDS(*qmgr-name*) CFSTRUCT(*struc-name*) now has ACCESS(*access*)

Severity

0

Explanation

The access availability setting for the shared message data set for the specified queue manager, and application structure has been changed to the indicated value, either by automatic status management or by a **RESET SMDS** command.

qmgr-name
identifies the queue manager, which owns the shared message data set.

struc-name
identifies the application structure associated with the shared message data set.

access
shows the new access availability setting. For details of specific settings, see the **DISPLAY CFSTATUS** command with the **TYPE(SMDS)** option.

System action

All queue managers connected to the structure are notified of the change. The queue managers take appropriate action if necessary, for example opening or closing the data set.

CSQE243I

SMDS(*qmgr-name*) CFSTRUCT(*struc-name*) now has DSBUFS(*value*)

Severity
0

Explanation

The number of shared message data set buffers to be used by the specified queue manager for this application structure has been changed to the indicated value. This message can either occur as a result of an **ALTER SMDS** command or when a previously specified **DSBUFS** target value cannot be achieved, in which case a warning message is issued, and the **DSBUFS** option is automatically set to the actual value achieved.

qmgr-name
identifies the queue manager, which owns the shared message data set.

struc-name
identifies the application structure associated with the shared message data set.

value
shows the new **DSBUFS** setting, which can either be a decimal number, giving the number of buffers to be used, or **DEFAULT**, indicating that the default **DSBUFS** value specified on the **CFSTRUCT** definition for the application structure is to be used. For more information, see the **ALTER SMDS** and **DISPLAY SMDS** commands.

System action

The queue manager identified by the **SMDS** keyword is notified, if active, and adjusts the size of its buffer pool as indicated.

CSQE244I

csect-name SMDS(*qmgr-name*) CFSTRUCT(*struc-name*) now has DSEXPAND(*value*)

Severity
0

Explanation

The option to allow automatic expansion of a specific shared message data set has been changed as indicated. This message can occur either as a result of an **ALTER SMDS** command or when expansion was attempted but failed, in which case the option is automatically changed to **DSEXPAND(NO)** to prevent further expansion attempts. In the latter case, when the problem has been fixed, the **ALTER SMDS** command can be used to turn automatic expansion on again.

qmgr-name
identifies the queue manager which owns the shared message data set.

struc-name
identifies the application structure associated with the shared message data set.

value
shows the new **DSEXPAND** setting, which is **DEFAULT**, **YES** or **NO**. For more information, see the **ALTER SMDS** and **DISPLAY SMDS** commands.

System action

The queue manager identified by the **SMDS** keyword is notified, if that queue manager is active. If the change results in expansion being enabled, and the data set is already in need of expansion, an immediate expansion is attempted.

CSQE245I

CFSTRUCT(*struc-name*) now has OFFLDUSE(*offload-usage*)

Severity
0

Explanation

The **OFFLOAD** method for an application structure was recently changed and the queue manager has now determined that there are no more messages stored using the old offload method, so there is no longer any need for the old offload method to remain active. The offload usage indicator, displayed as the **OFFLDUSE** keyword on the **DISPLAY CFSTATUS** command, has been updated to indicate that only the new offload method is now in use.

For a transition from **OFFLOAD(SMDS** to **OFFLOAD(DB2)**, this message occurs when all active data sets have been changed to the **EMPTY** state, which occurs if the data set is closed normally at a time when it does not contain any messages. In this case, the offload usage indicator is changed from **BOTH** to **DB2**, and the queue managers will no longer use the SMDS data sets, which can be deleted if no longer required.

For a transition from **OFFLOAD(DB2)** to **OFFLOAD(SMDS)**, this message occurs when the queue manager disconnects normally from the structure at a time when there are no large messages for the structure stored in Db2. In this case, the offload usage indicator is changed from **BOTH** to **SMDS**.

struc-name
identifies the application structure.

offload-usage
shows the new offload usage indicator.

System action

All queue managers connected to the structure are notified of the change. The queue managers take appropriate action if necessary, for example opening or closing data sets.

CSQE246I

csect-name SMDSCONN(*qmgr-name*) CFSTRUCT(*struc-name*) now has STATUS(*status*)

Severity
0

Explanation

The current queue manager was unable to connect to a shared message data set, usually for reasons indicated by a previous message. The error status for the data set connection has now been set to indicate the type of problem which occurred. It will be reset next time an attempt is made to open the data set.

This message is only issued for error status values, which are shown instead of normal status if the data set has been closed because of an error. No message is issued for normal status values (**CLOSED**, **OPENING**, **OPEN** or **CLOSING**).

qmgr-name
identifies the queue manager that owns the shared message data set.

struc-name
identifies the application structure associated with the shared message data set.

status
shows the new error status. For details of the possible status values, see the **STATUS** keyword on the **DISPLAY SMDSCONN** command.

System action

The **SMDSCONN** availability is set to **AVAIL(ERROR)** and message CSQE247I is issued.

No further attempt is made to connect to the data set until the availability value is changed back to **AVAIL(NORMAL)**. This can occur as a result of the queue manager being restarted, or data set availability changing, or in response to the **START SMDSCONN** command. If this happens while the queue manager is running, another message CSQE247I is issued showing **AVAIL(NORMAL)**.

CSQE247I

csect-name SMDSCONN(*qmgr-name*) CFSTRUCT(*struc-name*) now has AVAIL(*availability*)

Severity
0

Explanation

The availability setting for the connection between the current queue manager and a shared message data set has been changed to the indicated value. This can be changed either by automatic status management, for example if the queue manager is unable to open the data set, or by one of the commands **STOP SMDSCONN** or **START SMDSCONN**.

qmgr-name
identifies the queue manager that owns the shared message data set.

struc-name
identifies the application structure associated with the shared message data set.

availability
shows the new availability setting. For details of the possible values, see the **AVAIL** keyword on the **DISPLAY SMDSCONN** command.

System action

The current queue manager takes appropriate action if necessary, for example opening or closing the data set.

CSQE252I

SMDS(*qmgr-name*) CFSTRUCT(*struc-name*) data set *dsname* space map will be rebuilt by scanning the structure

Severity
0

Explanation

The data set space map needs to be reconstructed either following queue manager abnormal termination or data set recovery, so there will be a delay while this scan is completed.

qmgr-name
identifies the queue manager which owns the shared message data set.

struc-name
identifies the application structure associated with the shared message data set.

dsname
shows the full name of the shared message data set.

System action

The queue manager will scan the contents of the structure to determine which blocks in the data set are being referenced so that it can reconstruct the space map.

CSQE255I

SMDS(*qmgr-name*) CFSTRUCT(*struc-name*) data set *dsname* space map has been rebuilt, message count *msg-count*

Severity
0

Explanation

The scan to rebuild the data set space map has completed.

qmgr-name
identifies the queue manager which owns the shared message data set.

struc-name
identifies the application structure associated with the shared message data set.

dsname
shows the full name of the shared message data set.

msg-count
indicates the number of large messages currently stored in the data set.

System action

The data set is made available for use.

CSQE256E

SMDS(*qmgr-name*) CFSTRUCT(*struc-name*) data set *dsname* space map rebuild processing failed because a referenced message data block is beyond the end of the data set

Severity
8

Explanation

During the scan to rebuild the data set space map, a message was found in the structure which referenced a message data block with a control interval number greater than the size of the current data set. It is likely that the data set has been truncated.

qmgr-name
identifies the queue manager that owns the shared message data set.

struc-name
identifies the application structure associated with the shared message data set.

dsname
shows the full name of the shared message data set.

System action

The data set is closed and marked as **FAILED**.

System programmer response

This message indicates that the data set has been damaged, for example by copying it to a smaller data set, causing one or more message data blocks to be lost.

If the original copy is still available, the problem can be fixed without loss of data by reallocating the data set at the original size, copying in the original data, and then using the **RESET SMDS** command to mark the data set as **RECOVERED**.

Otherwise, any persistent messages can be recovered by recreating the data set at the original size and recovering the structure and the data set using the **RECOVER CFSTRUCT** command.

CSQE257E

SMDS(*qmgr-name*) CFSTRUCT(*struc-name*) data set *dsname* is smaller than the size recorded in the space map. The saved space map cannot be used

Severity
8

Explanation

The data set contained a saved space map, but the current size of the data set is smaller than the size recorded in the space map. It is likely that the data set has been truncated.

qmgr-name
identifies the queue manager that owns the shared message data set.

struc-name
identifies the application structure associated with the shared message data set.

dsname
shows the full name of the shared message data set.

System action

The saved space map is ignored and an attempt is made to rebuild the space map for the truncated data set. If all active message data is within the current extents of the data set the rebuild attempt will be successful, otherwise it will fail with message **CSQE256E**.

CSQE274E

The SMDS buffer pool for CFSTRUCT(*struc-name*) could not be created because insufficient storage was available

Severity
8

Explanation

Insufficient main storage was available to allocate the SMDS data buffer pool for the structure.

struc-name
identifies the application structure associated with the shared message data set.

System action

The data sets for this structure cannot be opened.

System programmer response

Consider increasing the queue manager's MEMLIMIT.

For more details about address space storage, see Address space storage.

CSQE275E

The SMDS buffer pool for CFSTRUCT(*struc-name*) has been created with *actual-buffers* rather than the requested *buffer-count* because insufficient storage was available

Severity
8

Explanation

Insufficient main storage was available to allocate the requested number of buffers in the SMDS data buffer pool for the structure. A smaller number of buffers were successfully allocated.

struc-name
identifies the application structure associated with the shared message data set.

actual-buffers
shows the number of buffers allocated.

buffer-count
shows the requested number of buffers.

System action

The buffer pool is created with a smaller number of buffers.

System programmer response

If the specified number of buffers is enough, change the requested value to match, to avoid similar problems in future.

Consider increasing the queue manager's MEMLIMIT.

For more details see Address space storage.

CSQE276I

The SMDS buffer pool for CFSTRUCT(*struc-name*) has been increased to *buffer-count* buffers

Severity
0

Explanation

The request to alter the **SMDS** buffer pool size has completed normally.

struc-name
identifies the application structure associated with the shared message data set.

buffer-count
shows the requested number of buffers.

System action

The additional buffers are made available for use.

CSQE277I

The SMDS buffer pool for CFSTRUCT(*struc-name*) has been increased to *actual-buffers* buffers rather than the requested *buffer-count* because insufficient storage was available

Severity
0

Explanation

The request to alter the **SMDS** buffer pool size has completed but the target number of buffers was not reached because insufficient main storage was available

struc-name
identifies the application structure associated with the shared message data set.

actual-buffers
shows the number of buffers allocated.

buffer-count
shows the requested number of buffers.

System action

The additional buffers are made available for use.

CSQE278I

The SMDS buffer pool for CFSTRUCT(*struc-name*) has been decreased to *buffer-count* buffers

Severity
0

Explanation

The request to reduce the **SMDS** buffer pool size has completed normally.

struc-name
identifies the application structure associated with the shared message data set.

buffer-count
shows the requested number of buffers.

System action

The storage for the excess buffers is released back to the system.

CSQE279I

The SMDS buffer pool for CFSTRUCT(*struc-name*) has been decreased to *actual-buffers* buffers rather than the requested *buffer-count* because the rest of the buffers are in use

Severity
0

Explanation

The request to reduce the **SMDS** buffer pool size could not reach the target number of buffers because the current number of buffers in use exceeded that number, and active buffers cannot be released.

struc-name
identifies the application structure associated with the shared message data set.

actual-buffers
shows the number of buffers allocated.

buffer-count
shows the requested number of buffers.

System action

If the number of buffers was at least partly reduced, the storage for the excess buffers is released back to the system.

CSQE280I

SMDS usage ...

Severity
0

Explanation

This message is issued in response to a **DISPLAY USAGE** command with **TYPE(SMDS)**. It shows the data set space usage information for the shared message data sets owned by the current queue manager for each application structure which is currently using SMDS support. The information is in the following format:

```
Application  Offloaded  Total  Total data  Used data  Used   Encr-
structure    messages   blocks  blocks      blocks    part    ypt
    n           n          n       n          n% ⋮  n
End of SMDS report
```

The columns of information are as follows:

Application structure
This is the name of the application structure.

Offloaded messages
This shows the number of shared messages in the structure for which the message data has been stored in the data set owned by this queue manager.

Total blocks
This is the current total size of the owned data set in logical blocks, including blocks used to store the space map.

Total data blocks
This is the number of blocks in the owned data set which can be used to store data, excluding those used to store the space map.

Used data blocks
This is the number of blocks in the owned data set which are currently in use (that is, one or more pages of those blocks contain active message data).

Used part
This is the ratio of the number of used data blocks to the total data blocks, expressed as a percentage.

Encrypt
This indicates whether the SMDS data set is encrypted (YES, or NO).

CSQE285I

SMDS buffer usage ...

Severity
0

Explanation

This message is issued in response to a **DISPLAY USAGE** command with **TYPE(SMDS)**. It shows the shared message data set buffer pool usage information for each application structure which is currently using SMDS support. The information is in the following format:

```
Application  Block  --------- Buffers ---------  Reads  Lowest  Wait
structure    size   Total  In use  Saved  Empty  saved  free   rate _name
nK          n          n          n          n         n%          n          n%
End of SMDS buffer report
```

The columns of information are as follows:

Application structure
This is the name of the application structure.

Block size
This shows the size of each buffer in Kbytes. This is equal to the logical block size of the shared message data set.

Buffers: Total
This is the actual number of buffers in the pool.

Buffers: In use
This is the number of buffers which are currently being used by requests to transfer data to or from the data set.

Buffers: Saved
This is the number of buffers which are free but currently contain saved data for recently accessed blocks.

Buffers: Empty
This is the number of buffers which are free and empty. When a new buffer is required, empty buffers are used first, but if there are no empty buffers, the least recently used saved buffer is reset to empty and used instead.

Reads saved
This is the percentage of read requests (during the current statistics interval) where the correct block was found in a saved buffer, avoiding the need to read the data from the data set.

Lowest free
This is the smallest number of free buffers during the current statistics interval, or zero if all buffers were used but no request had to wait for an empty buffer, or a negative number indicating the maximum number of requests which were waiting for a free buffer at the same time. If this value is negative, it indicates the number of additional buffers that would have been needed in order to avoid waits for a free buffer.

Wait rate
This is the fraction of requests to acquire a buffer which had to wait for a free buffer, expressed as a percentage. The numbers are reset when statistics are collected.

z/OS *Security manager messages (CSQH...)*

CSQH001I

Security using uppercase classes

Severity
0

Explanation

This message is issued to inform you that security is currently using the uppercase classes MQPROC, MQNLIST, MQQUEUE and MQADMIN.

CSQH002I

Security using mixed case classes

Severity
0

Explanation

This message is issued to inform you that security is currently using the mixed case classes MXPROC, MXNLIST, MXQUEUE and MXADMIN.

CSQH003I

Security refresh did not take place for class *class-name*

Severity
4

Explanation

This message follows message CSQH004I when an attempt to refresh class MQPROC, MQNLIST, or MQQUEUE was unsuccessful because of a return code from a SAF RACROUTE REQUEST=STAT call. The return code is given in message CSQH004I.

System action

The refresh does not occur.

System programmer response

Check that the class in question (*class-name*) is set up correctly. See message CSQH004I for the reason for the problem.

CSQH004I

csect-name STAT call failed for class *class-name*, SAF return code= *saf-rc*, ESM return code=*esm-rc*

Severity
8

Explanation

This message is issued as a result of a SAF RACROUTE REQUEST=STAT call to your external security manager (ESM) returning a non-zero return code at one of the following times:

- During initialization, or in response to a **REFRESH SECURITY** command

  If the return codes from SAF and your ESM are not zero, and are unexpected, this will cause abnormal termination with one of the following reason codes:

  – X'00C8000D'
  – X'00C80032'
  – X'00C80038'
- In response to a **REFRESH SECURITY** command.

  If the return codes from SAF and your ESM are not zero (for example, because a class is not active because you are not going to use it) this message is returned to the issuer of the command to advise that the STAT call failed.

Possible causes of this problem are:

- The class is not installed
- The class is not active
- The external security manager (ESM) is not active
- The RACF z/OS router table is incorrect

System programmer response

To determine if you need to take any action, see RACROUTE REQUEST=STAT (standard form) for more information about the return codes.

CSQH005I

csect-name resource-type In-storage profiles successfully listed

Severity
0

Explanation

This message is issued in response to a **REFRESH SECURITY** command that caused the in-storage profiles to be RACLISTED (that is, rebuilt); for example, when the security switch for a resource is set on, or a refresh for a specific class is requested that requires the in-storage tables to be rebuilt.

System programmer response

This message is issued so that you can check the security configuration of your queue manager.

CSQH006I

Error returned from CSQTTIME, security timer not started

Severity
8

Explanation

An error was returned from the MQ timer component, so the security timer was not started.

System action

The queue manager terminates abnormally, with a reason code of X'00C80042'.

System programmer response

See “Security manager codes (X'C8')” on page 958 for an explanation of the reason code.

CSQH007I

Reverify flag not set for user-id *userid*, no entry found

Severity
0

Explanation

A user identifier (*user-id*) specified in the RVERIFY SECURITY command was not valid because there was no entry found for it in the internal control table. This could be because the identifier was entered incorrectly in the command, or because it was not in the table (for example, because it had timed-out).

System action

The user identifier (*user-id*) is not flagged for reverify.

System programmer response

Check that the identifier was entered correctly.

CSQH008I

Subsystem security not active, no userids processed

Severity
0

Explanation

The RVERIFY SECURITY command was issued, but the subsystem security switch is off, so there are no internal control tables to flag for reverification.

CSQH009I

Errors occurred during security timeout processing

Severity
8

Explanation

This message is sent to the system log either:

- If an error occurs during security timeout processing (for example, a nonzero return code from the external security manager (ESM) during delete processing)
- Prior to a message CSQH010I if a nonzero return code is received from the timer (CSQTTIME) during an attempt to restart the security timer

System action

Processing continues.

System programmer response

Contact IBM Support to report the problem.

CSQH010I

csect-name Security timeout timer not restarted

Severity
8

Explanation

This message is issued to inform you that the security timeout timer is not operational. The reason for this depends on which of the following messages precedes this one:

CSQH009I
An error occurred during timeout processing

CSQH011I
The timeout interval has been set to zero

System action

If this message follows message CSQH009I, the queue manager ends abnormally with one of the following reason codes:

csect-name
Reason code

CSQHTPOP
X'00C80040'

CSQHPATC
X'00C80041'

System programmer response

See “Security manager codes (X'C8')” on page 958 for information about the reason code.

CSQH011I

csect-name Security interval is now set to zero

Severity
0

Explanation

The ALTER SECURITY command was entered with the INTERVAL attribute set to 0. This means that no user timeouts will occur.

System programmer response

This message is issued to warn you that no security timeouts will occur. Check that this is what was intended.

CSQH012I

Errors occurred during ALTER SECURITY timeout processing

Severity
8

Explanation

This message is issued in response to an ALTER SECURITY command if errors have been detected during timeout processing (for example, a nonzero return code from the external security manager (ESM) during timeout processing).

System action

Processing continues.

System programmer response

Contact IBM Support to report the problem.

CSQH013E

csect-name Case conflict for class *class-name*

Severity
8

Explanation

A **REFRESH SECURITY** command was issued, but the case currently in use for the class *class-name* differs from the system setting and if refreshed would result in the set of classes using different case settings.

System action

The refresh does not occur.

System programmer response

Check that the class in question (*class-name*) is set up correctly and that the system setting is correct. If a change in case setting is required, issue the **REFRESH SECURITY(*)** command to change all classes.

CSQH015I

Security timeout = *number* minutes

Severity
0

Explanation

This message is issued in response to the **DISPLAY SECURITY TIMEOUT** command, or as part of the **DISPLAY SECURITY ALL** command.

CSQH016I

Security interval = *number* minutes

Severity
0

Explanation

This message is issued in response to the **DISPLAY SECURITY INTERVAL** command, or as part of the **DISPLAY SECURITY ALL** command.

CSQH017I

Security refresh completed with errors in signoff

Severity
8

Explanation

This message is issued when an error has been detected in refresh processing; for example, a nonzero return code from the external security manager (ESM) during signoff or delete processing.

System action

Processing continues.

System programmer response

Contact IBM Support to report the problem.

CSQH018I

csect-name Security refresh for *resource-type* not processed, security switch set OFF

Severity
0

Explanation

A **REFRESH SECURITY** command was issued for resource type *resource-type*. However, the security switch for this type or the subsystem security switch is currently set off.

Note: This message is issued only for resource types MQQUEUE, MQPROC, and MQNLIST, because MQADMIN is always available for refresh.

System programmer response

Ensure that the **REFRESH SECURITY** request was issued for the correct resource type.

CSQH019I

Keyword values are incompatible

Severity
8

Explanation

The **REFRESH SECURITY** command was issued, but the command syntax is incorrect because a keyword value that is specified conflicts with the value for another keyword.

System action

The command is not executed.

System programmer response

See REFRESH SECURITY for more information.

CSQH021I

csect-name switch-type security switch set OFF, profile '*profile-type*' found

Severity
0

Explanation

This message is issued during queue manager initialization and in response to a **REFRESH SECURITY** command for each security switch that is set OFF because the named security profile has been found.

System action

If the subsystem security switch is set off, you will get only one message (for that switch).

System programmer response

Messages CSQH021I through CSQH026I are issued so that you can check the security configuration of your queue manager. See Switch profiles for information about setting security switches.

CSQH022I

csect-name switch-type security switch set ON, profile '*profile-type*' found

Severity
0

Explanation

This message is issued during queue manager initialization and in response to a **REFRESH SECURITY** command for each security switch that is set ON because the named security profile has been found.

System programmer response

Messages CSQH021I through CSQH026I are issued so that you can check the security configuration of your queue manager. See Switch profiles for information about setting security switches.

CSQH023I

csect-name switch-type security switch set OFF, profile '*profile-type*' not found

Severity
0

Explanation

This message is issued during queue manager initialization and in response to a **REFRESH SECURITY** command for each security switch that is set OFF because the named security profile has not been found.

System action

If the subsystem security switch is set off, you will get only one message (for that switch).

System programmer response

Messages CSQH021I through CSQH026I are issued so that you can check the security configuration of your queue manager. See Switch profiles for information about setting security switches.

CSQH024I

csect-name switch-type security switch set ON, profile '*profile-type*' not found

Severity
0

Explanation

This message is issued during queue manager initialization and in response to a **REFRESH SECURITY** command for each security switch that is set ON because the named security profile has not been found.

System programmer response

Messages CSQH021I through CSQH026I are issued so that you can check the security configuration of your queue manager. See Switch profiles for information about setting security switches.

CSQH025I

csect-name switch-type security switch set OFF, internal error

Severity
0

Explanation

This message is issued during queue manager initialization and in response to a **REFRESH SECURITY** command for each security switch that is set OFF because an error occurred.

System action

The message might be issued with message CSQH004I when an unexpected setting is encountered for a switch.

System programmer response

See message CSQH004I for more information.

Messages CSQH021I through CSQH026I are issued so that you can check the security configuration of your queue manager.

CSQH026I

csect-name switch-type security switch forced ON, profile '*profile-type*' overridden

Severity

0

Explanation

This message is issued during queue manager initialization and in response to a **REFRESH SECURITY** command for each security switch that was forced ON. This happens when an attempt was made to turn off both the queue manager and queue sharing group security switches for the named profile, which is not allowed.

System programmer response

Correct the profiles for the queue manager and queue sharing group security switches, and refresh security if required.

Messages CSQH021I through CSQH026I are issued so that you can check the security configuration of your queue manager. See Switch profiles for information about setting security switches.

CSQH030I

Security switches ...

Severity

0

Explanation

This is issued in response to a DISPLAY SECURITY ALL or DISPLAY SECURITY SWITCHES command and is followed by messages CSQH031I through CSQH036I for each security switch to show its setting and the security profile used to establish it.

System action

If the subsystem security switch is set off, you will get only one message (for that switch). Otherwise, a message is issued for each security switch.

CSQH031I

switch-type OFF, '*profile-type*' found

Severity

0

Explanation

This message is issued in response to a DISPLAY SECURITY ALL or DISPLAY SECURITY SWITCHES command for each security switch that is set OFF because the named security profile has been found.

System action

If the subsystem security switch is set off, you will get only one message (for that switch).

CSQH032I

switch-type ON, '*profile-type*' found

Severity
0

Explanation

This message is issued in response to a DISPLAY SECURITY ALL or DISPLAY SECURITY SWITCHES command for each security switch that is set ON because the named security profile has been found.

CSQH033I

switch-type OFF, '*profile-type*' not found

Severity
0

Explanation

This message is issued in response to a DISPLAY SECURITY ALL or DISPLAY SECURITY SWITCHES command for each security switch that is set OFF because the named security profile has not been found.

System action

If the subsystem security switch is set off, you will get only one message (for that switch).

CSQH034I

switch-type ON, '*profile-type*' not found

Severity
0

Explanation

This message is issued in response to a DISPLAY SECURITY ALL or DISPLAY SECURITY SWITCHES command for each security switch that is set ON because the named security profile has not been found.

CSQH035I

switch-type OFF, internal error

Severity
0

Explanation

This message is issued in response to a DISPLAY SECURITY ALL or DISPLAY SECURITY SWITCHES command for each security switch that is set OFF because an error occurred during initialization or when refreshing security.

System action

The message is be issued when an unexpected setting is encountered for a switch.

System programmer response

Check all your security switch settings. Review the z/OS system log file for other CSQH messages for errors during IBM MQ startup or when running RUNMQSC security refresh commands.

If required, correct them and refresh your security.

CSQH036I

switch-type ON, '*profile-type*' overridden

Severity
0

Explanation

This message is issued in response to a DISPLAY SECURITY ALL or DISPLAY SECURITY SWITCHES command for each security switch that was forced ON. This happens when an attempt was made to

turn off both the queue manager and queue sharing group security switches for the named profile, which is not allowed.

System programmer response

Correct the profiles for the queue manager and queue sharing group security switches, and refresh security if required.

CSQH037I

Security using uppercase classes

Severity
0

Explanation

This message is issued in response to a **DISPLAY SECURITY ALL** or **DISPLAY SECURITY SWITCHES** command to inform you that security is currently using the uppercase classes MQPROC, MQNLIST, MQQUEUE and MQADMIN.

CSQH038I

Security using mixed case classes

Severity
0

Explanation

This message is issued in response to a **DISPLAY SECURITY ALL** or **DISPLAY SECURITY SWITCHES** command to inform you that security is currently using the mixed case classes MXPROC, MXNLIST, MXQUEUE and MXADMIN.

CSQH040I

Connection authentication ...

Severity
0

Explanation

This message is issued during queue manager initialization, in response to a **DISPLAY SECURITY** command, and in response to a **REFRESH SECURITY TYPE(CONNAUTH)** command. It is followed by messages CSQH041I and CSQH042I to show the value of the connection authentication settings.

CSQH041I

Client checks: *check-client-value*

Severity
0

Explanation

This message is issued during queue manager initialization, in response to a **DISPLAY SECURITY** command, and in response to a **REFRESH SECURITY TYPE(CONNAUTH)** command. It shows the current value of connection authentication client checks.

If the value shown is '????' this means that the connection authentication settings were not able to be read. Preceding error messages will explain why. Any applications which connect while the queue manager is in this state will result in error message CSQH045E.

CSQH042I

Local bindings checks: *check-local-value*

Severity
0

Explanation

This message is issued during queue manager initialization, in response to a **DISPLAY SECURITY** command, and in response to a **REFRESH SECURITY TYPE(CONNAUTH)** command. It shows the current value of connection authentication local bindings checks.

If the value shown is '????' this means that the connection authentication settings were not able to be read. Preceding error messages will explain why. Any applications which connect while the queue manager is in this state will result in error message CSQH045E.

CSQH043E

csect-name Object AUTHINFO(*object-name*) does not exist or has wrong type

Severity

8

Explanation

During queue manager initialization or while processing a **REFRESH SECURITY TYPE(CONNAUTH)** command, the authentication information object named in the queue manager's CONNAUTH field was referenced. It was found to either not exist, or not have AUTHTYPE(IDPWOS).

System action

If this message is issued in response to a **REFRESH SECURITY TYPE(CONNAUTH)** command, the command fails and the connection authentication settings remain unchanged.

If this message is issued during queue manager initialization, all connection attempts are refused with reason “2035 (07F3) (RC2035): MQRC_NOT_AUTHORIZED” on page 1169 until the connection authentication settings have been corrected.

System programmer response

Ensure the authentication information object *object-name* has been defined correctly. Ensure the queue manager's CONNAUTH field is referencing the correct object name. Correct the configuration, then issue a REFRESH SECURITY TYPE(CONNAUTH) command for the changes to become active.

CSQH044E

csect-name Access to AUTHINFO(*object-name*) object failed, reason=*mqrc (mqrc-text)*

Severity

8

Explanation

During queue manager initialization or while processing a **REFRESH SECURITY TYPE(CONNAUTH)** command, the authentication information object named in the queue manager's CONNAUTH field could not be accessed for the reason given by *mqrc* (*mqrc-text* provides the MQRC in textual form).

System action

If this message is issued in response to a **REFRESH SECURITY TYPE(CONNAUTH)** command, the command fails and the connection authentication settings remain unchanged.

If this message is issued during queue manager initialization, all connection attempts are refused with reason “2035 (07F3) (RC2035): MQRC_NOT_AUTHORIZED” on page 1169 until the connection authentication settings have been corrected.

System programmer response

Ensure the authentication information object *object-name* has been defined correctly. Ensure the queue manager's CONNAUTH field is referencing the correct object name. Refer to “API completion and reason codes” on page 1151 for information about *mqrc* to determine why the object cannot be accessed. Correct the configuration, then issue a REFRESH SECURITY TYPE(CONNAUTH) command for the changes to become active.

CSQH045E

csect-name application did not provide a password

Severity

8

Explanation

An application connected without supplying a user ID and password for authentication and the queue manager is configured to require this type of application to supply one.

If this is a client application, the configuration attribute **CHCKCLNT** is set to REQUIRED. *application* is identified by *channel name*/*connection details*.

If this is a locally bound application, the configuration attribute **CHCKLOCL** is set to REQUIRED. *application* is identified by *user id*/*application name*.

If the connection authentication configuration was unable to be read, this message will also be seen. See messages CSQH041I and CSQH042I.

System action

The connection fails and the application is returned “2035 (07F3) (RC2035): MQRC_NOT_AUTHORIZED” on page 1169.

System programmer response

Ensure all applications are updated to supply a user ID and password, or alter the connection authentication configuration to OPTIONAL instead of REQUIRED, to allow applications to connect that have not supplied a user ID and password.

If the connection authentication configuration was unable to be read, check for earlier error messages and make corrections based on what is reported.

After making configuration changes, issue a REFRESH SECURITY TYPE(CONNAUTH) command for the changes to become active.

If the application is a client application, the user ID and password can be supplied without changing the application code, by using a security exit, such as mqccred, which is supplied with the IBM MQ MQI client.

CSQH046E

csect-name application supplied a password for user ID *userid* that has expired

Severity

8

Explanation

An application connected and supplied a user ID *userid* and password for authentication. The password supplied has expired.

If this is a client application, *application* is identified as 'channel name'/'connection details'.

If this is a locally bound application, *application* is identified as 'running user id'/'application name'.

System action

The connection fails and the application is returned “2035 (07F3) (RC2035): MQRC_NOT_AUTHORIZED” on page 1169.

System programmer response

Set a new password for *userid* using O/S facilities and retry the connect from the application using the new password.

z/OS *Data manager messages (CSQI...)*

CSQI002I

csect-name Page set *psid* value out of range

Severity
8

Explanation

One of the following commands has been issued:

- DEFINE STGCLASS
- DISPLAY STGCLASS
- DISPLAY USAGE

The value given for the page-set identifier was not in the range 0 through 99.

System action

The command is ignored.

System programmer response

Reissue the command using the correct syntax. (See MQSC commands for information about the command.)

CSQI003I

csect-name 'PSID' not allowed with TYPE (*usage-type*)

Severity
8

Explanation

A DISPLAY USAGE command was issued specifying both the PSID keyword and either TYPE(DATASET), or TYPE(SMDS), which is not allowed.

System action

The command is ignored.

System programmer response

Reissue the command using the correct syntax; see DISPLAY USAGE for additional information.

CSQI004I

csect-name Consider indexing *queue-name* by *index-type* for *connection-type* connection *connection-name*, *num-msgs* messages skipped

Severity
0

Explanation

The queue manager has detected an application receiving messages by message ID or correlation ID from a queue that does not have an index defined.

The type of index that should be established for the queue is indicated by *index-type*, and is either MSGID or CORRELID. The type of application that is affected is identified by *connection-type*, and is either BATCH, CHIN, CICS or IMS.

- For batch applications *connection-name* contains the job name.
- For the channel initiator *connection-name* contains the channel name.
- For CICS applications *connection-name* contains the region and transaction names.
- For IMS applications *connection-name* contains the IMS sysid, PSTID and PSB names.

The number of messages skipped while searching for the requested message, shown as *num-msgs*, is an indication of the impact of not having an index defined.

System action

Processing continues.

System programmer response

Investigate the application to determine whether an index is required for the queue.

The parameter to use with the DEFINE QLOCAL or ALTER QLOCAL command is **INDXTYPE**. Set it to *MSGID* or *CORRELID*, as indicated by the output you received for this message.

Applications that receive messages by message ID or correlation ID might encounter a performance degradation if an index is not defined and the depth of the queue is large.

CSQI005I

csect-name PAGE SET *nn* OFFLINE. RECOVERY RBA = *rba*

Severity
0

Explanation

This message indicates that the page set *nn* is currently not accessible by the queue manager. This might be because the page set has not been defined to the queue manager with the DEFINE PSID command.

This message can also be issued if the page set has been marked suspended.

Note: *rba* is the restart RBA for page set *nn*.

This situation can cause problems, so you should take action to correct it as soon as possible.

System action

Processing continues.

System programmer response

If the page set is required, bring it online; this can be done without stopping the queue manager. Use the FORMAT function of the utility program CSQUTIL, specifying TYPE(REPLACE). Then issue a DEFINE PSID command to bring the page set back into use. Note that all units of recovery (except those that are indoubt) that involved the offline page set will have been backed out by the queue manager when the page set was last used. These indoubt units of recovery may be resolved once the page set is back in use by the queue manager.

CSQI006I

csect-name COMPLETED IN-STORAGE INDEX FOR QUEUE *q-name*

Severity
0

Explanation

During restart, in-storage indexes are built for non-shared queues that have the INDXTYPE attribute, which might take some time. This message records that index-building has been completed for the specified queue.

System action

Processing continues.

CSQI007I

csect-name BUILDING IN-STORAGE INDEX FOR QUEUE *q-name*

Severity
0

Explanation

During restart, in-storage indexes are built for non-shared queues that have the INDXTYPE attribute, which might take some time. This message records that an index is being built for the specified queue.

System action

The in-storage index is built.

CSQI010I

Page set usage ...

Severity

0

Explanation

This message is the response to the DISPLAY USAGE command. It provides information about the page set usage, as follows:

```
  Page ...
  set
_  n   page-set-information ⋮
  End of page set report
```

where *n* is the page set identifier. The columns of *page-set-information* are:

Buffer pool
The buffer pool used by the page set.

Total pages
The total number of 4 KB pages in the page set (this relates to the records parameter on the VSAM definition of the page set).

Unused pages
The number of pages that are not used (that is, available page sets).

Persistent data pages
The number of pages holding persistent data (these pages are being used to store object definitions and persistent message data).

Nonpersistent data pages
The number of pages holding nonpersistent data (these pages are being used to store nonpersistent message data).

Expansion count
The type of expansion used for the page set (SYSTEM, USER, or NONE), and the number of times the page set has been dynamically expanded since restart. (The maximum number of times the page set can be expanded is constrained by the maximum number of extents allowable for the type of VSAM data set allocation and your operating system version.) If the count is large, your page set allocation might be wrong, or you might have some message processing problem.

Encrypt
The data set encryption status of the page set (YES, or NO).

Note: The page numbers are approximate because other threads might be altering the status of pages in this page set while the command is being processed.

If a page set is unavailable, *page-set-information* is one of:

has never been online
if the page set has been defined, but has never been used.

OFFLINE, recovery RBA=*rba*
if the page set is currently not accessible by the queue manager, for example because the page set has not been defined to the queue manager with the DEFINE PSID command; *rba* is the restart RBA for the page set.

is not defined
if the command was issued for a specific page set that is not defined to the queue manager.

is suspended, buffer pool *buffer pool number*, recovery RBA=*rba*
if the page set is suspended; *rba* is the restart RBA for the page set.

Exceptionally, the last line of the report might be:

```
Page set report terminated
```

if there was an error in obtaining the information. The error is described in the following messages.

CSQI012E

csect-name COULD NOT COMPLETE COMMAND. STORAGE EXHAUSTED

Severity
8

Explanation

A display of page set usage could not complete because all the available storage was exhausted.

System action

The output terminates at this point. There might be more information that has not been displayed. If this is in response to a DISPLAY USAGE command without the PSID keyword, try it again, specifying a page set identifier. This could decrease the amount of information produced, enabling it all to be displayed.

CSQI020I

MAXSMSGS(*number*)

Severity
0

Explanation

This message is issued in response to a DISPLAY MAXSMSGS command, and displays the maximum number of messages that a task can get or put within a single unit of recovery.

CSQI021I

csect-name PAGE SET *psid* IS EMPTY. MEDIA RECOVERY STARTED

Severity
0

Explanation

The queue manager has recognized a page set with a recovery RBA of zero. It will update the page set using information in the log data sets.

System action

The queue manager rebuilds the page set.

CSQI022I

csect-name PAGE SET *psid* NEWLY ADDED

Severity
0

Explanation

The queue manager has recognized that page set *psid* is new to the system.

CSQI023I

csect-name PAGE SET *psid* ONLINE AGAIN. MEDIA RECOVERY STARTED

Severity
0

Explanation

A page set has been redefined to the queue manager after a period offline or suspended.

System action

Any updates to the page set that are necessary are applied.

CSQI024I

csect-name Restart RBA for system as configured = *restart-rba*

Severity
0

Explanation

This message gives the restart RBA (relative byte address) for the queue manager, but does not include any offline or suspended page sets in the calculation of this restart point.

This value can be used to determine where to truncate logs, if you have no offline or suspended page sets.

If you have offline or suspended page sets that you want to add to your system at some time in the future, you must use the restart RBA given in message CSQI025I. If you truncate your logs at *rba* you might make it impossible to add the offline or suspended page sets back to the system.

CSQI025I

csect-name Restart RBA including offline page sets = *restart-rba*

Severity
0

Explanation

This message gives the restart RBA (relative byte address) for the queue manager, including any offline or suspended page sets.

This value can be used to determine where to truncate logs, if you have offline or suspended page sets that you want to add to the system in the future.

CSQI026I

csect-name PAGE SET *nn* DEFINED, BUT HAS NEVER BEEN ONLINE

Severity
0

Explanation

This message indicates that the page set *nn* has been defined, but it has never been used. Consequently, there is no restart RBA for the page set.

System action

Processing continues.

CSQI027I

csect-name PAGE SET *nn* TREATED AS A NEW PAGE SET

Severity
0

Explanation

This message indicates that the page set *nn* has been formatted using TYPE(NEW). It is treated as if it has been newly-added to the system, so all historical information relating to this page set is

discarded. In particular, all queues that use storage classes that reference the page set will be cleared of all messages.

System action

Processing continues.

CSQI028E

csect-name PAGE SET CONFLICT FOR QUEUE queue

Severity
8

Explanation

The named queue contains messages that are on a different page set from that associated with the storage class for the queue.

System action

This message might be issued more than once, each occurrence naming a different queue. The queue manager ends abnormally with reason code X'00C93800'.

System programmer response

Contact IBM Support for assistance.

CSQI029I

csect-name PAGE SET *psid* IS AN OLD COPY. MEDIA RECOVERY STARTED

Severity
0

Explanation

The queue manager has recognized that the media recovery RBA held within the page set is older than the media recovery RBA checkpointed for the page set. This is because the queue manager was started with an old copy of the page set.

System action

Any updates to the page set that are necessary are applied. Restart processing continues.

CSQI030I

csect-name PAGE SET *nn* TREATED AS A REPLACEMENT PAGE SET

Severity
0

Explanation

This message indicates that the page set *nn* has been formatted using TYPE(REPLACE). No media recovery will be performed on the page set.

System action

Processing continues.

CSQI031I

csect-name THE NEW EXTENT OF PAGE SET *psid* HAS FORMATTED SUCCESSFULLY

Severity
0

Explanation

Following the dynamic extension of page set *psid*, the new extent has been formatted successfully.

System action

Processing continues.

CSQI032I

csect-name NEW EXTENT(S) OF *nnn* PAGES DISCOVERED ON PAGE SET *psid* WILL NOW BE FORMATTED

Severity
0

Explanation

During restart, it was discovered that page set *psid* had been extended dynamically, but that *nnn* pages had not been formatted. This formatting will now be done.

System action

Processing continues.

CSQI033E

csect-name Block *block-number* of the message data for entry ID *entry-id* in CFSTRUCT(*struc-name*) was not found in Db2

Severity
8

Explanation

A shared message was read which referred to message data in Db2, but the corresponding data was not found in the Db2 table.

block-number
identifies the block number within the message of the data block which was not found.

entry-id
identifies the coupling facility entry for the shared message.

struc-name
identifies the application structure.

System action

If the message was persistent, the structure is marked as failed, requiring recovery, and messages CSQI036I and CSQE035E are issued.

If the message was nonpersistent, the damaged message is deleted and message CSQI037I is issued.

In both cases, a dump is produced.

CSQI034E

csect-name Block *block-number* of the message data for entry ID *entry-id* in CFSTRUCT(*struc-name*) refers to SMDS(*qmgr-id*) control interval *rci* but the stored data does not match the entry id

Severity
8

Explanation

A shared message was read which referred to message data stored in a shared message data set (SMDS), but when the data was read from the referenced location in the data set, the entry ID in the block prefix did not match the entry ID of the message.

block-number
identifies the block number within the message of the data block which was not found.

entry-id
identifies the coupling facility entry for the shared message.

struc-name
identifies the application structure.

qmgr-ide>
identifies the queue manager which owns the shared message data set.

rci
identifies the relative control interval number within the data set where the message block was expected to start.

System action

If the message was being retrieved for backup purposes, a dump is produced and the queue manager terminates.

Otherwise, action is taken as follows:

- If the message was persistent, the shared message data set and the structure are marked as failed, requiring recovery, and messages CSQI036I and CSQE035E are issued.
- If the message was nonpersistent, the damaged message is deleted and message CSQI037I is issued.

In both cases, a dump is produced.

CSQI035E

csect-name Block *block-number* of the message data for entry ID *entry-id* in CFSTRUCT(*struc-name*) refers to SMDS but the data set ID is not valid

Severity
8

Explanation

A shared message was read which referred to message data stored in a shared message data set (SMDS), but the relevant queue manager id (identified by the last byte of the entry id) is not one which currently owns a shared message data set.

block-number
identifies the block number within the message of the data block which could not be read.

entry-id
identifies the coupling facility entry for the shared message.

struc-name
identifies the application structure.

System action

If the message was persistent, the structure is marked as failed, requiring recovery, and messages CSQI036I and CSQE035E are issued.

If the message was nonpersistent, the damaged message is deleted and message CSQI037I is issued.

In both cases, a dump is produced.

CSQI036I

csect-name CFSTRUCT(*struc-name*) has been marked as failed because the data for persistent message with entry ID *entry-id* could not be retrieved

Severity
0

Explanation

A damaged persistent message was found, so the structure has been marked as failed, requiring recovery.

struc-name
identifies the application structure.

entry-id
identifies the coupling facility entry for the shared message.

System action

The structure is marked as failed and message CSQE035E is issued.

CSQI037I

csect-name The nonpersistent message with entry ID *entry-id* has been deleted from CFSTRUCT(*struc-name*) because the data could not be retrieved

Severity
0

Explanation

A damaged nonpersistent message was found which could not be successfully retrieved, so it has been deleted.

entry-id
identifies the coupling facility entry for the shared message.

struc-name
identifies the application structure.

System action

The damaged message is deleted. No attempt is made to delete any associated SMDS message data.

CSQI038I

csect-name The damaged message with entry id *entry-id* in CFSTRUCT(*struct-name*) is for queue *queue-name*

Severity
0

Explanation

A damaged shared message entry has been found, as indicated by a previous message, and this message indicates the corresponding queue name.

struc-name
identifies the application structure.

entry-id
identifies the coupling facility entry for the shared message.

queue-name
identifies the queue for which the message cannot be retrieved.

System action

Processing continues. This message will be followed by message CSQI036I or CSQI037I, depending on whether the damaged message was persistent or not.

CSQI039E

csect-name LRSN required for structure recovery not available for one or more CF structures

Explanation

The LRSN required for structure recovery for one or more CF structures could not be located within the logs indexed in the BSDS.

Previous CSQE040I and CSQE041E messages might indicate which CF structures are causing this error to occur.

System action

Processing continues.

System programmer response

Use the **BACKUP CFSTRUCT** command, on any queue manager in the queue sharing group, to make a new CF structure backup. You might consider setting up a procedure to take frequent backups automatically.

CSQI041I

csect-name JOB *jobname* USER *userid* HAD ERROR ACCESSING PAGE SET *psid*

Severity
0

Explanation

This message is issued when there is an error on a page set. The message identifies the job name, user ID, and page set identifier associated with the error.

CSQI042E

csect-name WLM IWMCONN request failed, rc=*rc* reason=*reason*

Severity
8

Explanation

A Workload Management Services (WLM) connect call failed. *rc* is the return code and *reason* is the reason code (both in hexadecimal) from the call.

System action

Processing continues, but WLM services are not available.

System programmer response

See the *z/OS MVS Programming: Workload Management Services* manual for information about the return and reason codes from the WLM call. When you have resolved the problem, you will need to restart the queue manager. If you are unable to solve the problem, contact IBM Support for assistance.

CSQI043E

csect-name WLM *call-name* request for process *process-name* failed, rc=*rc* reason=*reason*

Severity
8

Explanation

A Workload Management Services (WLM) call failed. *rc* is the return code and *reason* is the reason code (both in hexadecimal) from the call.

System action

Processing continues, but WLM services are not available.

System programmer response

See the *z/OS MVS Programming: Workload Management Services* manual for information about the return and reason codes from the WLM call. When you have resolved the problem, you will need to restart the queue manager. If you are unable to solve the problem, contact IBM Support for assistance.

CSQI044I

csect-name Process *process-name* used by queue *q-name* was not found

Severity
0

Explanation

The named queue is indexed by message tokens. An action was being performed for the queue that required the use of the Workload Management Services (WLM) IWMCLSFY service. However, the process specified by the queue does not exist, so the service name for WLM cannot be determined.

System action

A blank service name is passed to the Workload Management Services (WLM) IWMCLSFY service.

System programmer response

Correct the queue or process definitions.

CSQI045I

csect-name Log RBA has reached *rba*. Plan a log reset

Severity
4

Explanation

The current log RBA is approaching the end of the log RBA.

System action

Processing continues, unless the RBA value reaches FFF800000000 (if 6-byte log RBAs are in use) or FFFFFFC000000000 (if 8-byte log RBAs are in use) when the queue manager terminates with reason code 00D10257.

System programmer response

Plan to stop the queue manager at a convenient time and reset the logs. See RESETPAGE for information on how to reset the logs using the CSQUTIL utility program and resetting the queue manager's log.

If your queue manager is using 6-byte log RBAs, consider converting the queue manager to use 8-byte log RBAs. See Planning to increase the maximum addressable log range for further information.

CSQI046E

csect-name Log RBA has reached *rba*. Perform a log reset

Severity
8

Explanation

The current log RBA is approaching the end of the log RBA.

System action

Processing continues, unless the RBA value reaches FFF800000000 (if 6-byte log RBAs are in use) or FFFFFFC000000000 (if 8-byte log RBAs are in use) when the queue manager terminates with reason code 00D10257.

System programmer response

Stop the queue manager as soon as it is convenient and reset the logs. See RESETPAGE for information on how to reset the logs using the CSQUTIL utility program and resetting the queue manager's log.

If your queue manager is using 6-byte log RBAs, consider converting the queue manager to use 8-byte log RBAs. See Planning to increase the maximum addressable log range for further information.

CSQI047E

csect-name Log RBA has reached *rba*. Stop queue manager and reset logs

Severity
8

Explanation

The current log RBA is too close to the end of the log RBA range.

System action

Processing continues, unless the RBA value reaches FFF800000000 (if 6-byte log RBAs are in use) or FFFFFFC000000000 (if 8-byte log RBAs are in use) when the queue manager terminates with reason code 00D10257.

System programmer response

Stop the queue manager immediately and reset the logs. See RESETPAGE for information on how to reset the logs using the CSQUTIL utility program and resetting the queue manager's log.

If your queue manager is using 6-byte log RBAs, consider converting the queue manager to use 8-byte log RBAs. See Planning to increase the maximum addressable log range for further information.

CSQI048I

csect-name WLM reached maximum enclave limit

Severity
4

Explanation

Workload Management Services (WLM) reported that no more enclaves could be created, so a message could not be notified to WLM. (An IWMECREA call gave a return code of 8 with a reason code of X'xxxx0836'.)

Note: This message might be issued repeatedly during the scan of the indexes for WLM-managed queues.

System action

The queue manager will attempt to notify the message to WLM again on the next scan of the indexes for WLM-managed queues. This will be after the interval specified by the WLMTIME system parameter. For information about the system parameters for the CSQ6SYSP macro, see Using CSQ6SYSP.

System programmer response

See the *z/OS MVS Programming: Workload Management Services* manual for information about the return and reason codes from the WLM call.

CSQI049I

Page set *psid* has media recovery RBA=*rcvry-rba*, checkpoint RBA= *chkpt-rba*

Severity
0

Explanation

During restart, the queue manager opened the indicated page set. The media recovery RBA from the page set itself and the check pointed RBA from the logs are as shown.

If the RBAs differ, it indicates that an old copy of the page set is being used. If the checkpoint RBA and the prior checkpoint RBA shown in message CSQR003I differ, it indicates that the page set has been offline or suspended.

System action

Processing continues. Media recovery is performed if necessary to bring the page set up to date.

CSQI050E

csect-name Page set *psid value* RESETPAGE needed

Severity
8

Explanation

The queue manager has detected that a page set contains RBA values higher than the maximum logged.

System action

During queue manager startup, restart is terminated abnormally with completion code X'5C6' and reason code X'00C94525'.

During DEFINE PSID command processing, the command fails.

System programmer response
Run CSQUTIL with the RESETPAGE utility against the page set or sets indicated in the messages and retry the failed operation.

CSQI051E

csect-name QDEPTHHI less than QDEPTHLO for queue *queue*

Severity
8

Explanation

At start up a queue was found to have QDEPTHHI set to a value less than the value of QDEPTHLO.

System action

Processing continues.

System programmer response
Correct the queue definition so that QDEPTHHI is greater than or equal to QDEPTHLO.

CSQI052E

Invalid spacemap RBA found during restart for page set *psid*

Severity
8

Explanation

A space map page containing an invalid RBA was detected on the indicated page set during startup, indicating the page set is not in a consistent state.

This is normally as a result of the page set not being correctly processed during a past cold start operation or RESETPAGE operation.

System action

The page set is suspended. Queues using the page set will be inaccessible until the queue manager is started with the page set in a consistent state.

System programmer response

When *psid* specifies page set 0, contact IBM Service.

For page sets other than 0, plan to stop the queue manager as soon as it is convenient, then follow the procedure to restore the page set or sets to a consistent state:

- Run CSQUTIL with SCOPY PSID(x) to save persistent messages on the page set to a data set
- Format the page set with TYPE(NEW)
- Start the queue manager and reload the messages from the data set using SCOPY LOAD

CSQI053E

Invalid page RBA found during restart for page set *psid*

Severity
8

Explanation

A page containing an invalid RBA was detected on the indicated page set during startup, indicating the page set is not in a consistent state.

This is normally as a result of the page set not being correctly processed during a past cold start operation or RESETPAGE operation.

System action

The page set is suspended. Queues using the page set will be inaccessible until the queue manager is started with the page set in a consistent state. If the page set id is 0 then during queue manager startup, restart is terminated abnormally with completion code X'5C6' and reason code X'00C94531'.

System programmer response

When *psid* specifies page set 0, contact IBM Service.

For page sets other than 0, plan to stop the queue manager as soon as it is convenient, then follow the procedure to restore the page set or sets to a consistent state:

- Run CSQUTIL with SCOPY PSID(x) to save persistent messages on the page set to a data set
- Format the page set with TYPE(NEW)
- Start the queue manager and reload the messages from the data set using SCOPY LOAD

CSQI059E

Unable to increase cluster cache

Severity
8

Explanation

The dynamic cluster cache cannot be increased because the queue manager cluster cache task encountered an error.

System action

The cluster cache task terminates. The channel initiator will probably terminate.

System programmer response

Investigate the problem reported in any preceding messages.

CSQI060E

QSG names differ, log=*log-name* queue manager=*qmgr-name*

Severity
8

Explanation

The queue sharing group name recorded in the log does not match the name being used by the queue manager.

Possible causes are:

- The queue manager was restarted using the log from another queue manager.
- The queue manager was restarted with the wrong QSGDATA system parameter.
- The queue manager was not removed correctly from its previous queue sharing group.

System action

Restart is terminated abnormally with completion code X'5C6' and reason code X'00C94505'.

System programmer response

Restart the queue manager using the correct logs and BSDS, or change the QSGDATA system parameter. Note that you cannot change the name of the queue sharing group that a queue manager

uses, or remove it from a queue sharing group, unless it has been shut down normally and the further procedures for removal described in Managing queue sharing groups have been followed.

CSQI061E

Queue manager queue sharing group numbers differ, log=*log-num* queue manager=*qmgr-num*

Severity
8

Explanation

The queue manager was restarted using the log from another queue manager. The queue sharing group queue manager number recorded in the log does not match that being used by the queue manager.

System action

Restart is terminated abnormally with completion code X'5C6' and reason code X'00C94506'.

System programmer response

Restart the queue manager using the correct logs and BSDS. If the correct logs are being used, correct the entry for the queue manager in the Db2 CSQ.ADMIN_B_QMGR table. If you cannot resolve the problem, contact IBM Support for assistance.

CSQI062I

Queue *q-name* deleted by another queue manager during restart

Severity
0

Explanation

During restart processing the queue manager detected that the named queue has been deleted by another queue manager in the queue sharing group.

System action

Processing continues.

CSQI063E

Queue *q-name* is both PRIVATE and SHARED

Severity
0

Explanation

During restart processing the queue manager detected that the named queue exists both as a locally-defined queue on this queue manager and as a shared queue in the queue sharing group. Opening a queue with this name will therefore not be allowed.

System action

Processing continues.

System programmer response

Delete one of the instances of the queue. See Shared queue problems for more information.

CSQI064E

Cannot get information from Db2. *obj-type* COPY objects not refreshed

Severity
8

Explanation

During queue manager or channel initiator startup, objects of type *obj-type* with a disposition of COPY were being refreshed from those with a disposition of GROUP. However, the necessary information could not be obtained from Db2; this may be because Db2 is not available or no longer available, or because the connection to Db2 is suspended, or because there was an error in accessing Db2, or because a Db2 table was temporarily locked.

System action

The COPY objects of type *obj-type* are not refreshed. Startup continues.

System programmer response

Refer to the console log for messages giving more information about the error.

When the error condition has cleared, refresh the objects manually, or restart the queue manager or channel initiator.

CSQI065I

Buffer pool attributes ...

Severity

0

Explanation

This message displays the current state of buffer pool attributes, based on the page set number passed into the **DISPLAY USAGE PSID** command. It provides information about the number of available buffers, buffers free (stealable), shown as a number and as a percentage of the buffers in the pool, and the memory LOCATION for the specified buffer pool.

```
CSQI065I !MQ21 Buffer pool attributes ...  321
  Buffer  Available  Stealable   Stealable  Page      Location
    pool    buffers    buffers  percentage  class
_      0       5000       4989          99  FIXED4KB  ABOVE
_      1       5000       4995          99  4KB       ABOVE
_      2       5000       4999          99  4KB       BELOW
_      3       5000       4995          99  4KB       BELOW
_      4       5000       4999          99  4KB       BELOW
_      5       1000        999          99  4KB       BELOW
```

Buffer pool

The number of the buffer pool.

Available buffers

The total number of available buffers defined for a specified buffer pool.

If location is SWITCHING_ABOVE or SWITCHING_BELOW, the value is the sum of the numbers above and below.

Stealable buffers

The number of buffers free (stealable) for a defined buffer pool.

Stealable percentage

The amount of buffers free (stealable), as a percentage, for a defined buffer pool.

Page class

The type of virtual storage pages used for backing the buffers in the buffer pool. The page class value is one of the following:

4KB

Buffers are backed by standard pageable 4 KB pages

FIXED4KB

Buffers are backed by permanently page-fixed 4 KB page

Location

The location value of the memory used by individual buffer pools. The location value is one of the following:

ABOVE
Memory is used above the bar for buffer pools.

BELOW
BELOW is the default. Memory is used below the bar for buffer pools.

SWITCHING_ABOVE
The buffer pool is in the process of switching to a location ABOVE the bar.

SWITCHING_BELOW
The buffer pool is in the process of switching to a location BELOW the bar.

CSQI070I

Data set usage ...

Severity
0

Explanation

This message is the response to the DISPLAY USAGE command. It provides information about the data sets relating to various circumstances, as follows:

```
Data set  RBA/LRSN  DSName
data-set-type:
       rrr       dsname
End of data set report
```

where:

data-set-type
The type of data set and circumstance, which can be:

Log, oldest with active unit of work
The log data set containing the beginning RBA of the oldest active unit of work for the queue manager.

Log, oldest for page set recovery
The log data set containing the oldest restart RBA of any page set for the queue manager.

Log, oldest for CF structure recovery
The log data set containing the LRSN which matches the time of the oldest current backup of any CF structure in the queue sharing group. If the oldest current backup is not found, you must back up all of your structures.

rrr
The RBA or LRSN corresponding to the circumstance.

dsname
The name of the copy 1 data set. If no data set relates to a circumstance, this is shown as `None`; if the data set name cannot be determined, this is shown as `Not found`.

System programmer response

This information can be used to help manage data sets; see Tips for backup and recovery for more information.

CSQI090E
RRS is not available.

Severity
8

Explanation
RRS was called to register interest on behalf of an IBM MQ application linked with an RRS stub, but RRS is not available.

System action
The IBM application is terminated with completion code X'5C6' and reason code X'00C94201'. This message is issued unless it has been issued recently

System programmer response
Ensure RRS is available when using an IBM MQ application linked with the RRS stub.

CSQI965I
modulename Backward migration required for msgs on page set *ps-name*

Explanation
During queue manager restart it has been detected that one or more of the page sets that have been connected has been used at a higher version of queue manager code.

System action

The queue manager will automatically perform special processing during restart to alter any messages stored on the indicated page set so they can be read by the current version of the queue manager.

CSQI968I
modulename Alias queue *aq-name* to TARGQ *tq-name* has TARGTYPE *ttype* which is not supported. *aq-name* has been deleted

Explanation
During object migration, an alias queue was found which had an invalid **TARGTYPE**, for example an alias queue to a topic object.

System action

The alias queue indicated is deleted.

CSQI969I
Data set *ds-name* for page set *ps-name* was used for a higher version of IBM MQ and cannot be added dynamically

Explanation

During dynamic connection to a page set which was offline at queue manager restart, it has been detected that it requires backward migration processing.

The page set is not dynamically added.

CSQI970E
csect-name object-type(*object-name*) COULD NOT BE MIGRATED

Explanation

Migration of the identified object could not be performed because of locks held by in-doubt transactions.

Some functions will not be available until migration of the object can be performed. For example, the object cannot be altered or deleted, and if it is a transmission queue, the associated channel may not start.

System action

The object is not migrated.

System programmer response

Use the DISPLAY CONN or the DISPLAY THREAD command to identify the list of in-doubt transactions and then resolve them via either the transaction coordinator or the RESOLVE INDOUBT command. Once the in-doubt transactions are resolved, either restart the queue manager or issue an ALTER command against the object to re-attempt its migration.

Message CSQI971I will be issued when the object has been successfully migrated.

CSQI971I
csect-name object-type(*object-name*) MIGRATED

Explanation

The identified object could not be migrated when the queue manager was first started at the current version because of locks held by in-doubt transactions (see message CSQI970E for more information).

This message is issued during a subsequent restart of the queue manager, or when the object is subsequently altered, to indicate that migration of the object has now occurred.

System action

The object is migrated.

System programmer response

none.

z/OS *Recovery log manager messages (CSQJ...)*

CSQJ001I
CURRENT COPY *n* ACTIVE LOG DATA SET IS DSNAME=*dsname*, STARTRBA=*sss* ENDRBA=*ttt*

Explanation

This message is generated for one of two reasons:

1. When the queue manager starts, this information message is sent to identify the current active log data sets (copy 1 and, if dual logging is used, copy 2).
2. When the current active log data set is full (or when an **ARCHIVE LOG** command is issued), IBM MQ will switch to the next available active log data set. This message identifies the next available active log data set that will be used for logging.

The value specified by **STARTRBA** is the RBA of the first byte of log data in the named data set. The value specified by **ENDRBA** is the RBA of the last possible byte in the data set.

System programmer response

None required. However, if recovery is required, information from this message might be required as input to the change log inventory utility (CSQJU003).

CSQJ002I
END OF ACTIVE LOG DATA SET DSNAME=*dsname*, STARTRBA=*sss* ENDRBA=*ttt*

Explanation

This message is sent when logging switches to a new empty data set. The message shows the name and log RBA range of the full data set.

System programmer response

None required. However, if recovery is required, information from this message might be required as input to the change log inventory utility (CSQJU003).

CSQJ003I
FULL ARCHIVE LOG VOLUME DSNAME=*dsname*, STARTRBA=*sss* ENDRBA=*ttt*, STARTTIME=*ppp* ENDTIME=*qqq*, UNIT=*unitname*, COPY*n*VOL=*vvv* VOLSPAN=*xxx* CATLG=*yyy*

Explanation

Offloading for the specified archive log data set was successfully completed for the given volume. If the data set spans multiple tape volumes, this message is generated for each tape volume.

System action

An archive log data set has been created, and the archive log data set inventory in the BSDS has been updated with the information in the message:

DSNAME
The name of the archive log data set

STARTRBA
The starting RBA contained in the volume

ENDRBA
The ending RBA contained in the volume

STARTTIME
The starting store-clock value of the log records in the volume

ENDTIME
The ending store-clock value of the log records in the volume

UNIT
The device unit to which the data set was allocated

COPY*n*VOL
The name of the volume; this is displayed as COPY1VOL if this is the copy-1 archive log data set, and as COPY2VOL if this is the copy-2 archive log data set

VOLSPAN
An indicator to denote one of four conditions:

NO
The data set is entirely contained on the volume specified by COPYnVOL

FIRST
This is the first entry of a multivolume data set

MIDDLE
This is the middle entry of a multivolume data set

LAST
This is the last entry of a multivolume data set

CATLG
An indicator to denote one of two conditions:

NO
The archive log data set is uncataloged

YES
The archive log data set is cataloged

The BSDS is automatically updated with the information contained in this message; however, if recovery is required, information from this message might be required as input to the change log inventory utility (CSQJU003).

CSQJ004I
ACTIVE LOG COPY *n* INACTIVE, LOG IN SINGLE MODE, ENDRBA=*ttt*

Explanation

This message is sent when the dual active logging option is selected and copy *n* becomes inactive. A log copy becomes inactive when the next active log data set is not ready when required. **ENDRBA** is the last byte of log data written on copy *n*. This is usually caused by a delay in offload.

System action

The log is switched to single mode until the next data set for copy *n* is ready for logging.

If the queue manager is shut down or terminates abnormally while in single mode with the system parameter option still set for dual active data sets, the previous state of the active log data sets determines what happens when the queue manager is started, as follows:

- If fewer than two data sets are available (not flagged as STOPPED) for each set of active logs, queue manager startup terminates and message CSQJ112E is issued.
- If an active log data set is in NOTREUSABLE state, the queue manager can be started in single logging mode, but dual mode takes effect when the other active log data set becomes available after offloading.

System programmer response

Perform a display request to ensure that there are no outstanding requests that are related to the log offload process. Take the necessary action to satisfy any requests, and permit offload to continue.

If the switch to single mode was caused by the lack of a resource required for offload, the necessary resource should be made available to allow offload to complete and thus permit dual logging to proceed. If recovery is required, information from this message might be required as input to the change log inventory utility (CSQJU003).

CSQJ005I
ACTIVE LOG COPY *n* IS ACTIVE, LOG IN DUAL MODE, STARTRBA=*sss*

Explanation

This message is sent when copy *n* of the log becomes active after previously being flagged as inactive. STARTRBA is the RBA of the first byte of log data written on copy *n* after it was activated.

System programmer response

None required. However, if recovery is required, information from this message might be required as input to the change log inventory utility (CSQJU003).

CSQJ006I
ALLOCATION FOR NEW ARCHIVE LOG DATA SET HAS BEEN CANCELED BY OPERATOR

Explanation

This message is sent if the operator answers 'N' to message CSQJ008E.

System action

If the allocation is for the first copy of the archive log data set, offload terminates processing until the next time it is activated. If the first copy has already been allocated and this request is for the second copy, offload switches to single offload mode for this data set only.

CSQJ007I
ALLOCATION FOR ARCHIVE VOL SER=*volser* HAS BEEN CANCELED BY OPERATOR

Explanation

If the operator answers 'N' to message CSQJ009E, this message is issued. *volser* is the volume serial of an archive log volume required to satisfy the read request. The name of the archive data set is given by message CSQJ022I which follows.

System action

The read request that needed the archive volume is unsuccessful. If the request was issued with the *COND=YES* parameter, the log manager returns to its invoker with return code 12 and reason code X'00D1032B'. Otherwise, the log manager's invoker ends abnormally with the same reason code.

CSQJ008E
nn OF *mm* ACTIVE LOGS ARE FULL, *qmgr-name* NEEDS ARCHIVE SCRATCH

Explanation

IBM MQ needs a scratch volume for offloading an active log data set. *qmgr-name* is the name of the queue manager. *nn* is the number of full active log data sets. *mm* is the total number of active log data sets.

System action

The offload task issues message CSQJ021D and waits for the operator's reply.

CSQJ009E
qmgr-name NEEDS VOL SER= *nnnnnn*

Explanation

MQ needs the specified archive volume for a read operation. *qmgr-name* is the name of the queue manager.

System action

The archive log read service task issues message CSQJ021D and waits for the operator's reply. This wait affects the agent for which the log read was issued and any other agents that might be waiting on the log read service task queue.

CSQJ010I
INVALID RESPONSE - NOT Y OR N

Explanation

During archive data set allocation, a reply message was issued. The user did not respond correctly to the reply message. Either 'Y' or 'N' must be entered.

System action

The original message is repeated.

CSQJ011D
RESTART CONTROL *rrr* CREATED AT *date time* FOUND. REPLY Y TO USE, N TO CANCEL

Explanation

During queue manager initialization, a conditional restart control record was found in the BSDS data set. Both the record identifier (a 4-byte hexadecimal number) and the creation time stamp are displayed to help identify the conditional restart record which will be used. If you want a conditional restart using that record, reply 'Y' to the message. Otherwise, reply 'N'.

System action

If 'Y' is the response, the queue manager is started conditionally, using the record found. If 'N' is the response, startup is terminated.

System programmer response

Respond as indicated.

If a normal restart has failed and you have created a conditional restart record with the change log inventory utility, check whether the time and date in the message agree with when you created that record. If they do, reply 'Y'. If they do not, reply 'N' and investigate the discrepancy.

CSQJ012E
ERROR *ccc* READING RBA *rrr* IN DATA SET *dsname*, CONNECTION-ID=*xxxx* THREAD-XREF=*yyyyyy*

Explanation

While scanning log records read into a buffer, IBM MQ detected a logical error with reason code *ccc*. *rrr* is the log RBA of the segment in the buffer at which the error was detected. *dsname* is the name of the active or archive log data set from which the record was read. If *dsname* is blank, the data was read from an active log output buffer.

The connection ID and thread-xref identify the user or application that encountered the problem. Messages that have the same connection ID and thread-xref relate to the same user.

System action

The application program is terminated with reason code *ccc*. However, information in this message might be useful in diagnosing the abnormal termination that will follow.

System programmer response

See Active log problems for information about dealing with problems on the log.

CSQJ013E
TERMINAL ERROR *ccc* IN BUFFER *rrr* BEFORE ACTIVE LOG WRITE

Explanation

A scan of the log output buffer, just prior to writing the buffer, detected an inconsistency in the log data. *ccc* is the reason code associated with the SDUMP that is produced. *rrr* is the log RBA at which the error was detected.

System action

The queue manager will terminate with a dump, and will not write the damaged buffer to either COPY 1 or COPY 2 active log data set.

System programmer response

Restart the queue manager after it terminates.

Because the damaged buffer has not been written to a log data set, the queue manager can be restarted. No corrective action is required.

CSQJ014E
TERMINAL ERROR *ccc* IN BUFFER *rrr* AFTER ACTIVE LOG WRITE

Explanation

A scan of the log output buffer, after writing to the first copy of the active log data set and before writing to the second copy, detected an inconsistency in the log data. *ccc* is the reason code associated with the SDUMP that is produced. *rrr* is the log RBA at which the error was detected.

System action

The queue manager terminates with a dump, and does not write the damaged buffer to the COPY 2 data set.

System programmer response

The block containing the indicated log RBA might be damaged. The buffer was found to be in error at the completion of the write to the COPY 1 data set of the active log.

If dual active logs are being used, use the print log map utility (CSQJU004) to list the active log data sets for both copies of the active log. Find the COPY 2 data set with the corresponding RBA, and copy that data set (using Access Method Services REPRO) to the COPY 1 data set. Start the queue manager.

If only a single active log is used, contact the IBM support center for assistance. An attempt to start the queue manager might succeed if the damage to the buffer occurred after completion of the write to DASD.

CSQJ020I
csect-name RECEIVED REPLY OF N TO *msg-num*. QUEUE MANAGER STARTUP IS TERMINATED

Explanation

The operator chose to terminate queue manager startup by answering 'N' to *msg-num*.

System action

The queue manager will not restart.

CSQJ021D
REPLY Y WHEN DEVICE READY OR N TO CANCEL

Explanation

An archive log data set needs allocating, as indicated in the preceding CSQJ008E or CSQJ009E message.

System action

The log service task waits for the operator's reply.

CSQJ022I
DSNAME=*dsname*

Explanation

dsname is the name of the archive data set to which the preceding message refers.

CSQJ030E
RBA RANGE *startrba* TO *endrba* NOT AVAILABLE IN ACTIVE LOG DATA SETS

Explanation

Previous errors have made the active log data sets (that contain the RBA range reported in the message) unavailable. The status of these logs is STOPPED in the BSDS.

System action

The queue manager terminates with a dump.

System programmer response

The log RBA range must be available for the queue manager to be recoverable. Correct the previous errors and restore the active log data sets that contain the RBA range reported in the message.

- If the log data sets are recoverable, the active log data set inventory in the BSDS must be modified to reset the STOPPED status. Use the print log map utility (CSQJU004) to obtain a copy of the BSDS log inventory. Next, use the change log inventory utility (CSQJU003) to delete the active log data sets marked STOPPED (use the DELETE statement), then add them again (use the NEWLOG statement). The starting and ending RBA for each active log data set must be specified on the NEWLOG statement when the logs are added back to the BSDS using the change log inventory utility.
- If the log data sets are not recoverable, see Active log problems for information about dealing with problems on the log.

CSQJ031D
csect-name, THE LOG RBA RANGE MUST BE RESET. REPLY 'Y' TO CONTINUE STARTUP OR 'N' TO SHUTDOWN.

Explanation

If, during queue manager initialization, the current log RBA value is equal or higher than FF8000000000 (if 6-byte log RBAs are in use) or FFFFFC0000000000 (if 8-byte log RBAs are in use) this message is issued for the operator to confirm if the restart of the queue manager should continue.

System action

If 'Y' is the response, the queue manager startup continues.

If 'N' is the response, the queue manager startup terminates.

System programmer response

Stop the queue manager and reset the logs as soon as possible. See RESETPAGE for information on how to reset the logs using the CSQUTIL utility program and resetting the queue manager's log.

If your queue manager is using 6-byte log RBAs, consider converting the queue manager to use 8-byte log RBAs. See Planning to increase the maximum addressable log range for further information.

CSQJ032E
csect-name alert-lvl - APPROACHING END OF THE LOG RBA RANGE OF *max-rba*. CURRENT LOG RBA IS *current-rba*.

Explanation

The current log RBA is approaching the end of the log RBA range. *current-rba* is the current log RBA value. The current log RBA should not be allowed to advance to the maximum log RBA value of *max-rba*.

This message is issued during queue manager initialization, or after the active log data set is full and the queue manager switches to the next available log data set.

`alert-lvl` indicates one of the following:

WARNING
Issued when the current log RBA reaches the F80000000000 value (if 6-byte log RBAs are in use) or FFFFC00000000000 (if 8-byte log RBAs are in use).

CRITICAL
Issued after the log RBA value reaches FF8000000000 (if 6-byte log RBAs are in use) or FFFFFC0000000000 (if 8-byte log RBAs are in use).

System action

Processing continues, unless the RBA value reaches FFF800000000 (if 6-byte log RBAs are in use) or FFFFFFC000000000 (if 8-byte log RBAs are in use) when the queue manager terminates with reason code 00D10257.

System programmer response

Plan to stop the queue manager and reset the logs as soon as possible; see RESETPAGE for information on how to reset the logs using the CSQUTIL utility program and resetting the queue manager's log.

If your queue manager is using 6-byte log RBAs, consider converting the queue manager to use 8-byte log RBAs. See Planning to increase the maximum addressable log range for further information.

CSQJ033I
FULL ARCHIVE LOG VOLUME DSNAME=*dsname*, STARTRBA= *sss* ENDRBA=*ttt*, STARTLRSN=*ppp* ENDLRSN=*qqq*, UNIT=*unitname*, COPYnVOL=*vvv* VOLSPAN=*xxx* CATLG=*yyy*

Explanation

Offloading for the specified archive log data set was successfully completed for the given volume. If the data set spans multiple tape volumes, this message is generated for each tape volume.

This message is issued in place of CSQJ003I for queue sharing groups.

System action

See message CSQJ003I. **STARTTIME** and **ENDTIME** are replaced by the following:

STARTLRSN
The starting LRSN contained in the volume for queue sharing groups.

ENDLRSN
The ending LRSN contained in the volume for queue sharing groups.

CSQJ034I
csect-name END OF LOG RBA RANGE IS *max-rba*

Explanation

This message is issued during queue manager startup, to indicate the end of the log RBA range that can be addressed using the current log RBA size.

A *max-rba* value of 0000FFFFFFFFFFFF indicates that the queue manager is configured to use 6 byte RBAs, while a value of FFFFFFFFFFFFFFFF indicates that the queue manager is configured to use 8 byte RBAs.

You must reset the queue manager's log before the highest used log RBA reaches the end of the log RBA range.

System action

Processing continues

System programmer response

If *max-rba* is 0000FFFFFFFFFFFF, consider converting the queue manager to use an 8-byte log RBA, to maximize the period of time until a reset of the queue manager's log is required. See Planning to increase the maximum addressable log range for further information.

CSQJ060E
parm-name system parameters are unusable

Explanation

The format of the parameters set by *parm-name* in the system parameter load module is invalid, so they cannot be used.

System action

The queue manager is terminated with abnormally with reason code X'00E80084'.

System programmer response

Ensure that the queue manager is started with a correct system parameter module, for example CSQZPARM. If necessary, reassemble the module that uses the indicated parameters, and relink-edit your system parameter load module.

CSQJ061I
parm-name system parameters are obsolete

Explanation

The parameters set by *parm-name* in the system parameter load module use some values which are now obsolete.

System action

Processing continues. The obsolete parameters are ignored, and default values are used for new parameters.

System programmer response

Review your system parameter settings. If necessary, reassemble the module that uses the indicated parameters, and relink-edit your system parameter load module.

CSQJ070E
csect-name ARCHIVE LOG DSN PREFIX NOT IN PROPER FORMAT TO RECEIVE TIME STAMP DATA. TIME STAMPING OF *dsname* BYPASSED

Explanation

The system parameters (set by CSQ6ARVP) specify that the date and time of creation of an archive log data set be included as part of the archive log data set name (DSN). To accomplish this, IBM MQ requires that the length of the archive log data set name prefix is limited. If the prefix requirement is not met, this message is issued just prior to the allocation of the archive log data set specified in the message.

System action

The archive log data set will be allocated using the archive log prefix. However, the archive log DSN will not contain the date and time as the user requested.

System programmer response

The system parameters for the log archive function must be changed. Specifically, the **TSTAMP** and **ARCPFXn** fields are not consistent with one another. For information about the actions required to eliminate this problem, see Using CSQ6ARVP.

CSQJ071E
csect-name TIMER FAILURE CAUSED TIME STAMPING OF ARCHIVE *dsname* TO BE BYPASSED

Explanation

The system parameters (set by CSQ6ARVP) specify that the date and time of creation of an archive log data set be included as part of the archive log data set name (DSN). However, an attempt to get the current date and time from the system was unsuccessful. This message is issued just prior to the allocation of the archive log data set specified in the message.

System action

The archive log data set will be allocated using the archive log prefix. However, the archive log DSN will not contain the date and time as the user requested.

CSQJ072E
ARCHIVE LOG DATA SET *dsname* HAS BEEN ALLOCATED TO NON-TAPE DEVICE AND CATALOGED, OVERRIDING CATALOG PARAMETER

Explanation

The system parameters (set by CSQ6ARVP) specify that all archive log data sets should be uncataloged (CATALOG=NO). However, MQ requires that all archive log data sets allocated to non-tape devices must be cataloged. The archive log data set specified by *dsname* has been allocated to a non-tape device, and has thus been cataloged. The user's system parameter CATALOG setting of NO has been overridden.

System action

The archive log data set has been allocated to a nontape device, and has been cataloged. The system parameter CATALOG=NO setting has been overridden. The BSDS reflects that the data set has been cataloged.

System programmer response

The archive system parameters must be changed. Specifically, the **CATALOG** and **UNIT** parameters are not consistent with one another. For information about the actions required to eliminate this problem, see Using CSQ6ARVP.

CSQJ073E
LOG ARCHIVE UNIT ALLOCATION FAILED, REASON CODE= *ccc*. ALLOCATION OR OFFLOAD OF ARCHIVE LOG DATA SET MAY FAIL

Explanation

While building the SVC99 text entries to allocate a new archive log data set dynamically, a unit allocation error was detected. The reason code, indicated by *ccc* in the message, further clarifies the problem as follows:

4-28 (X'4'-X'1C')
Return code from z/OS IEFGB4UV macro. Common values are:

4 (X'04')
Invalid unit name

8 (X'08')
Unit name has incorrect units assigned

16 (X'10')
No storage available

20 (X'14')
Device numbers not valid

32 (X'20')
IBM MQ was able to obtain a list of devices corresponding to the device type (unit name) specified in the system parameters. However, it was determined that this list contained a mixture of tape and nontape devices.

36 (X'24')
Nonfetch-protected storage could not be obtained to build a parameter list for a z/OS service.

40 (X'28')
The device type (unit name) specified by the user in the system parameters is valid. However, no devices are currently associated with the given device type (unit name).

44 (X'2C')
The device type (unit name) specified by the user in the system parameters is valid. However, no DASD volumes are available with a volume use attribute of *storage*.

System action

This message is issued after the SVC99 text entries are built, but prior to the allocation of the new archive log data set. As a result of the error, the dynamic allocation of the archive log data set will

be attempted using standard default values. The standard default values are generally acceptable; however, the allocation might be unsuccessful or the subsequent offload might produce undesirable processing results. For example:

- A reason code of 4 or 44 (X'2C') indicates an allocation error (CSQJ103E) when the SVC99 is issued for the archive data set.
- Offload processing to tape might be unsuccessful. IBM MQ uses a volume count of 20 when allocating to tape, and uses the standard z/OS volume count default of 5 volumes when writing to non-tape devices. In the case of most of the above errors, it would be impossible for IBM MQ to determine the device type on which the data set is to be allocated. Therefore, the standard z/OS default is assumed for the volume count. If the data set is successfully allocated to a tape device, and the volume of data is such that more than five volumes will be used for the archive data set, the offload processing will receive a z/OS completion code X'837-08' with message IEC028I when attempting to write to the sixth tape volume.
- Offload processing to a direct access device might be unsuccessful. When allocating a new archive log data set on a direct access device, IBM MQ will use a unit count to facilitate multivolume archive data sets. With most of the above errors, it might be impossible for IBM MQ to correctly determine the type of device on which the data set is to be allocated. Therefore, the standard default (1) is assumed for the unit count. If the data set is successfully allocated to a direct access device, and during the offload processing it becomes necessary to extend the data set to another device, the offload processing will receive a z/OS X'B37' (out of space) completion code, and the archive log data set will be deallocated.

System programmer response

The required action is based on the reason code indicated in the message:

4-28 (X'4'-X'1C')
See the *z/OS MVS Programming: Authorized Assembler Services Guide* for more info about the return code from the z/OS IEFGB4UV macro. The most likely causes for the common values are:

4 (X'04')
Incorrect specification in the archive system parameters. Correct the UNIT parameter. If the UNIT parameter from the archive system parameters appears to be correct, check the EDT to ensure that the esoteric or generic unit name specified in the parameters is actually in the EDT. Subsequent offload processing will archive the log data which could not be previously archived due to the allocation error (CSQJ103E).

8 (X'08')
Incorrect specification in archive system parameters, incorrect operational setup.

16 (X'10')
This is usually a temporary problem. If the allocation of the archive log data set is successful, no action is required to correct this situation. If this is a recurring problem, sufficient page space is not available, and the region size for the queue manager address space might have to be increased, or standard z/OS diagnostic procedures might have to be used to correct the problem.

20 (X'14')
Incorrect specification in archive system parameters, incorrect operational

32 (X'20') or 40 (X'28')
To correct this situation, change the archive system parameter UNIT to use a device type (unit name) that contains homogenous devices, or modify the device list associated with the device type (unit name) using a system generation to supply a list of homogenous devices.

44 (X'2C')
To correct this situation, issue the z/OS command **MOUNT** to change the volume use attribute of a mounted private volume to storage. If this is a recurring problem, you might have to do one of the following:

- Perform a system generation to add permanently resident volumes with a volume use attribute of storage to the esoteric or generic unit

- Change the archive system parameters to use a different esoteric or generic unit name for the UNIT

CSQJ077E
LOG OR BSDS READ ERROR FOR QMGR *qmgr-name*, REASON CODE=*ccc*

Explanation

This message identifies a queue manager with log data that cannot be accessed. The logs or BSDSs of other queue managers in a queue sharing group might be accessed during a **RECOVER CFSTRUCT** operation or during the rebuild of peer administration structures that might occur on a queue manager in a queue sharing group.

System action

The execution unit writes a record to SYS1.LOGREC and requests an SVC dump. The execution unit then terminates abnormally.

System programmer response

Look for earlier messages which might identify more specifically the data set being accessed and the problem.

For reason code:

- 00D10901, which indicates that the peer SCA member record of the peer IBM MQ cannot be found, you can resolve the problem by restarting the peer queue manager.
- 00D10905, which indicates that there is a mismatch between the two BSDS timestamps of the peer queue manager, you can resolve the problem by restarting the peer queue manager.

The **REASON CODE** might also come from Db2. If this is the case, see the Db2 reason codes for more information.

If you are unable to solve the problem, note the reason code, collect the following items, and contact IBM Support:

- System dump
- Console output for the issuing queue manager
- Console output for the other queue manager
- Printout of SYS1.LOGREC

CSQJ098E
csect-name RESTART CONTROL ENDLRSN *rrr* IS NOT IN KNOWN LRSN RANGE. QUEUE MANAGER STARTUP IS TERMINATED

Explanation

A conditional restart control record requests truncation, but it cannot take place because the end LRSN was not in the range of LRSN values known to either the active or archive logs. *rrr* is the end LRSN specified in the active record. The end LRSN is either higher than the end LRSN of the most recent active log data set, or lower than the starting LRSN of the oldest archive log data set.

System action

Queue manager startup is terminated.

System programmer response

Check the **ENDLRSN** value specified in the conditional restart control record. If it is not correct, run the change log inventory utility (CSQJU003) using CRESTART CANCEL cancel the conditional restart, and a new CRESTART specifying the correct **ENDLRSN**.

CSQJ099I
LOG RECORDING TO COMMENCE WITH STARTRBA= *sss*

Explanation

This message is generated during queue manager startup. The value specified by **STARTRBA** is the RBA of the next byte of log data to be recorded in the active log data sets.

This message is preceded by one (if single logging) or two (if dual logging) CSQJ001I messages.

System programmer response

None required. However, if recovery is required, information from this message might be required as input to the change log inventory utility (CSQJU003).

CSQJ100E
csect-name ERROR OPENING BSDS*n* DSNAME= *dsname*, ERROR STATUS=*ee,ii*

Explanation

During startup, or while processing a **RECOVER BSDS** command, IBM MQ could not open the specified BSDS. BSDS*n* matches the DDname in the queue manager started task JCL procedure (xxxxMSTR) of the data set that cannot be opened. The value of *n* is 1 or 2. The error status contains the VSAM open return code in *ee*, and the VSAM open reason code in *ii*.

System action

When this error occurs at initialization time, startup must be terminated, because the log data sets cannot be determined and allocated without the BSDS. When this error occurs during **RECOVER BSDS** processing, the command is terminated, and the queue manager continues in single BSDS mode.

System programmer response

Recover the BSDS that cannot be opened. See Active log problems for information about dealing with problems on the BSDS or the log.

CSQJ101E
csect-name RESTART CONTROL ENDRBA *rrr* IS NOT IN KNOWN RBA RANGE. QUEUE MANAGER STARTUP IS TERMINATED

Explanation

A conditional restart control record requests truncation, but it cannot take place because the end RBA was not in the range of RBA values known to either the active or archive logs. *rrr* is the end RBA specified in the active record. The end RBA is either higher than the end RBA of the most recent active log data set, or lower than the starting RBA of the oldest archive log data set.

System action

Queue manager startup is terminated.

System programmer response

Check the **ENDRBA** value specified in the conditional restart control record. If it is not correct, run the change log inventory utility (CSQJU003) using CRESTART CANCEL cancel the conditional restart, and a new CRESTART specifying the correct **ENDRBA**.

Otherwise, then most likely, the archive log data set that contained the requested RBA has been deleted from the BSDS data set by the change log inventory utility. Locate the output from an old print log map utility and identify the data set that contains the missing RBA. If the data set has not been reused, run the change log inventory utility to add this data set back into the inventory of log data sets. Restart the queue manager.

CSQJ102E
LOG RBA CONTENT OF LOG DATA SET DSNAME= *dsname*, STARTRBA= *sss* ENDRBA=*ttt*, DOES NOT AGREE WITH BSDS INFORMATION

Explanation

The log RBA range shown in the BSDS for the specified data set does not agree with the content of the data set.

System action

Startup processing is terminated.

System programmer response

Use the print log map and change log inventory utilities to make the BSDS consistent with the log data sets.

CSQJ103E

csect-name LOG ALLOCATION ERROR DSNAME=*dsname*, ERROR STATUS=*eeeeiiii*, SMS REASON CODE=*ssssssss*

Explanation

An error occurred while attempting to allocate the active or archive log data set indicated by **DSNAME**. **STATUS** indicates the error reason code returned by z/OS dynamic allocation (SVC99).

This message might be preceded by message CSQJ073E.

System action

Subsequent actions depend on the type of data set involved.

For active log data sets, if the error is encountered during queue manager initialization, startup is terminated. If two copies of the active log data sets are defined, this message appears only once.

For archive log data sets, if two copies of the archive log data sets are defined, processing continues on the remaining archive log data set.

System programmer response

The error status portion of this message contains a 2-byte error code (*eeee*, S99ERROR) followed by the 2-byte information code (*iiii*, S99INFO) from the SVC99 request block. If the S99ERROR code indicates an SMS allocation error ('97xx'), then *ssssssss* contains additional SMS reason code information obtained from S99ERSN.

Go to the *z/OS MVS Programming: Authorized Assembler Services Guide* and select the Interpreting DYNALLOC return codes topic for information about these codes .

For active log data sets, if the problem occurred during queue manager initialization, you can resolve the problem by doing one of the following:

- Resolve the error associated with the active log data set as indicated by STATUS
- Provide another copy of the active log data set, using Access Method Services
- Update the BSDS with the change log inventory utility (CSQJU003)
- Restart the queue manager

For archive log data sets:

- If the problem occurred during allocation with the intent to write the data set, no immediate action is required. However, if you do not resolve the SVC99 error (indicated by the STATUS value in the message), the available space in the active log could eventually be exhausted (CSQJ111A) because all future offloads might be unsuccessful because of the same error.
- If the problem occurred during allocation with the intent to read the data set, determine the problem, and use the change log inventory utility (CSQJU003) DELETE function to delete the archive log data set from the BSDS archive log inventory. Then use the NEWLOG function to add the data set back into the archive log inventory, pointing to the correct volume and device.

See Active log problems for information about dealing with problems on the log.

This message might also be issued as the result of a user error. If STATUS displays a value of '17080000', you might have one or more active log data sets defined in the BSDS, but not allocated on DASD. To correct the situation, print the contents of the current active log data set inventory using the print log map utility (CSQJU004), then either:

- Use Access Method Services to allocate the active log data set for each active log data set listed in the BSDS, but not actually allocated on DASD. You can find the Access Method Services command syntax for active log data sets in the CSQ4BSDS sample JCL.
- Use the change log inventory utility (CSQJU003) DELETE statement to delete the errant active log data set name, and the NEWLOG statement to add the correct name to the active log data set inventory. The name specified on the NEWLOG statement must be the same as the name of the actual active log data set allocated on DASD.

CSQJ104E
csect-name RECEIVED ERROR STATUS *nnn* FROM *macro-name* FOR DSNAME *dsname*

Explanation

An error occurred while issuing macro *macro-name*. Error status is the return code from the specified macro:

- For an OPEN of a VSAM data set, the return code in the error field of the Access Method Services control block is included in this message as the error status value. See the *z/OS DFSMS Macro Instructions for Data Sets* manual for a description of these values.
- If the OPEN was for a non-VSAM data set, the error status is zero.
- For MMSRV errors, error status contains the error information returned by media manager services. If an MMSRV CATUPDT error occurs attempting to truncate an active log data set, the log data set will be unavailable and the status of the log data set will be flagged as STOPPED in the BSDS.
- For VSAM OPEN and MMSRV errors, this message is preceded by an IEC161I message that defines the error that occurred.
- For a PROTECT of an archive log data set, the return code is from DADSM PROTECT.

See Active log problems for information about dealing with problems on the log.

System action

Subsequent actions depend on when the error occurred.

During queue manager initialization, startup is terminated.

When using the data set either for offload or for input operations, processing continues. If a second copy of the data is available, IBM MQ attempts to allocate and open the second data set.

When using the data set as an active log data set, IBM MQ attempts to retry the request. If the retry is unsuccessful, the queue manager is terminated.

During checkpoint processing, where IBM MQ attempts to locate the oldest active or archive log data sets that are required for restart recovery of page sets and restart and media recovery of CF structures, processing continues. The message is a warning that either restart recovery would fail or media recovery of CF structures would fail. It is most likely to occur when all CF application structures are not being regularly backed up, thereby requiring excessively old log data sets for recovery.

System programmer response

If the error occurred during initialization, either correct the problem so that the data set is available or provide another copy of the data set and change the BSDSs to point to the new data set.

If the error occurred after startup, the return code should be reviewed and the appropriate action taken to correct the problem, so that the data set can be used at a later time, or the data set entry can be removed from the BSDS using the change log inventory utility.

If the error was received from PROTECT, there might be a problem with the PASSWORD data set. See the appropriate DADSM publication to determine the cause of the problem. When the problem has been corrected, ensure the archive log data sets receiving the error are added to the PASSWORD data set. If these archive log data sets are not added to the PASSWORD data set, archive read will not be able to OPEN these data sets. If you do not have information about the named macro, note the macro name and the return code and contact IBM Support for help.

If the error occurred during checkpoint processing, issue the **DISPLAY USAGE TYPE(DATASET)** command to show which log data sets are currently required for page set and media recovery, and ensure that they are available. If applicable, use the **BACKUP CFSTRUCT** command for your CF structures, and institute a procedure to back up your CF structures frequently.

CSQJ105E

csect-name LOG WRITE ERROR DSNAME= *dsname*, LOGRBA=*rrr*, ERROR STATUS=*ccccffss*

Explanation

An error occurred while writing a log data set. If *csect-name* is CSQJW107, the error occurred writing the log buffers to an active log data set. If *csect-name* is CSQJW207, the error occurred while pre-formatting the next control area before writing log data into it.

Error status contains the error information returned by media manager in the form *ccccffss*, where *cccc* is a 2-byte return code that describes the error, *ff* is a 1-byte code that defines the functional routine that detected the error, and *ss* is the 1-byte status code that defines a general category of error.

System action

If the dual active logging option is selected, then IBM MQ switches to the next data set for this copy. If the next data set is not ready, IBM MQ temporarily enters single logging mode and allocates a replacement data set for the one that encountered the error. Dual logging is resumed as soon as possible.

If single active logging option is selected and the next data set is not ready, IBM MQ waits for that data set to be available. In this case, log writing is inhibited until the replacement is ready for output.

System programmer response

See the *z/OS DFSMSdfp Diagnosis* manual for information about Media manager return codes. If you are unable to resolve the problem, note the return code, and contact IBM Support.

CSQJ106E

LOG READ ERROR DSNAME=*dsname*, LOGRBA=*rrr*, ERROR STATUS=*ccccffss*

Explanation

An error occurred while reading an active log data set. The error status contains the error information returned by the media manager in the form *ccccffss*, where *cccc* is a 2-byte return code that describes the error, *ff* is a 1-byte code that defines the functional routine that detected the error, and *ss* is the 1-byte status code that defines a general category of error. (See the *z/OS DFSMSdfp Diagnosis* manual for information about Media manager return codes.)

System action

If another log data set contains the data, IBM MQ attempts to read the data from the alternate source. If an alternate source is not available, a read error return code is sent to the program requesting the log data. Depending on the circumstances under which the failure occurred, the queue manager might continue with the alternate log data set if dual logging is used, or end abnormally.

System programmer response

If you are using dual logging, the requested RBA was probably retrieved from the corresponding dual active log data set, and no immediate response is necessary. However, if this error occurs frequently, or if you are using single logging, immediate attention might be required. If so, note the contents of the error status field, and contact IBM support center for help.

It might be necessary to replace the data set in error with a new data set containing the log data, and to update the BSDSs to reflect the new data set using the change log inventory (CSQJU003) NEWLOG operation.

See Active log problems for information about dealing with problems on the log.

This message might also be issued as the result of a user error. If the data set name specified by DSNAME is missing, and STATUS displays a value of '00180408' or '00100408', you are using dual

logging, but only one set of active log data sets is defined in the BSDS. To resolve this condition, do either of the following:

- Define a second set of active log data sets using Access Method Services (if they are not defined already), and update the BSDS log inventory, using the change log inventory (CSQJU003) NEWLOG operation.
- Reset the log system parameters to indicate single logging. You can do this by setting **TWOACTV** to 'NO' in the CSQ6LOGP system parameters.

CSQJ107E
READ ERROR ON BSDS DSNAME=*dsname* ERROR STATUS=*eee*

Explanation

An error occurred while reading the specified BSDS. Error Status contains the VSAM return and feedback codes. It is a 2-byte field with the first byte containing the hexadecimal return code and the second containing the hexadecimal feedback code. See the *z/OS DFSMS Macro Instructions for Data Sets* manual for a description of VSAM return and reason codes.

See Active log problems for information about dealing with problems on the BSDS or the log.

System action

If dual BSDSs are available, IBM MQ attempts to read from the other BSDSs. If the read from the second BSDS fails or if there is only one BSDS, an error code is returned to the log request that caused access to the BSDS.

If the read error is detected during startup, the queue manager terminates.

If the error code is unexpected, a diagnostic dump might be taken.

System programmer response

It might be necessary to replace or repair the BSDS, depending on what conditions resulted from the read error. To replace a BSDS, first delete the BSDS in error, then define the new BSDS with the same name and attributes. If a new name is used for the new BSDS, change the queue manager started task JCL procedure (xxxxMSTR) to specify the new BSDS name.

CSQJ108E
WRITE ERROR ON BSDS DSNAME=*dsname* ERROR STATUS=*eee*

Explanation

An error occurred while writing to the specified BSDS. Error Status contains the VSAM return and feedback codes. It is a 2-byte field with the first containing the hexadecimal return code and the second containing the hexadecimal feedback code. See the *z/OS DFSMS Macro Instructions for Data Sets* manual for a description of VSAM return and reason codes.

System action

If dual BSDSs are available, IBM MQ enters single BSDS mode using the remaining good BSDS. Otherwise, an error code is returned to the log request that caused access to the BSDS.

If the error code is unexpected, a diagnostic dump might be taken.

System programmer response

If dual BSDS mode is being used, run an offline Access Method Services job to rename the error BSDS and define a new BSDS with the same name. Then enter the RECOVER BSDS command to reestablish dual BSDS mode.

If dual BSDS mode is not being used, the queue manager must be shut down, and the BSDS must be recovered from a backup copy. To recover the BSDS, use the change log inventory utility.

CSQJ109E
OUT OF SPACE IN BSDS DSNAME=*dsname*

Explanation

There is no more space in the specified BSDS. The operation that encountered the out-of-space condition did not complete properly.

System action

If dual BSDSs are available, IBM MQ enters single BSDS mode using the remaining good BSDS. Otherwise, an error code is returned to the log request that caused access to the BSDS.

System programmer response

If dual BSDS mode is being used, run an offline Access Method Services job to rename the full BSDS and define a new, larger BSDS with the same name. Enter the RECOVER BSDS command to reestablish dual BSDS mode.

If dual BSDS mode is not being used, the queue manager must be shut down and the BSDS recovered offline. In this case, run the same Access Method Services job mentioned above to rename the full data set and define a larger data set. Next, run an Access Method Services REPRO job to copy the full BSDS into the new BSDS.

CSQJ110E
LAST COPY*n* ACTIVE LOG DATA SET IS *nnn* PERCENT FULL

Explanation

This message is issued when the last available active log data set is 5% full, and is reissued after each additional 5% of the data set space is filled.

System action

Each time the message is issued, the offload processing will be re-attempted. If the situation is not corrected, the active log data set will fill to capacity, message CSQJ111A will be issued, and IBM MQ processing will stop.

System programmer response

To clear this condition, you must take steps to complete other waiting offload tasks. Once an active log data set is made available (reusable) by completing the offload process for it, the IBM MQ logging activity can continue.

Perform a display request to determine the outstanding requests related to the log offload process. Take the necessary action to satisfy any requests, and permit offload to continue.

Consider whether there are sufficient active log data sets. If necessary, additional log data sets can be added dynamically using the **DEFINE LOG** command.

If offload does not complete normally or cannot be initiated, either correct the problem that is causing the offload process error, increase the size of the allocated data sets, or add more active log data sets. Note that the latter action requires the queue manager to be inactive and the change log inventory utility to be run.

Possible causes for the shortage of active log data space are:

- Excessive logging. For example, there is a lot of persistent message activity.
- Delayed or slow offloading. For example, failure to mount archive volumes, incorrect replies to offload messages, or slow device speeds.
- Excessive use of the **ARCHIVE LOG** command. Each invocation of this command causes IBM MQ to switch to a new active log data set and to initiate an offload of the active log. Although the command will not be processed when only one active log data set remains in a copy of the active log (see CSQJ319I), excessive use of the command could have consumed all space in the active log except the current active log data sets.
- Offloads were unsuccessful.
- Insufficient active log space.

CSQJ111A
OUT OF SPACE IN ACTIVE LOG DATA SETS

Explanation

Due to delays in offload processing, all available space in all active log data sets has been exhausted. Recovery logging cannot continue.

System action

IBM MQ waits for an available data set. Any tasks performing IBM MQ API calls that require logging will wait.

System programmer response

Perform a display request to ensure that there are no outstanding requests that are related to the log offload process. Take the necessary action to satisfy any requests, and permit offload to continue.

Consider whether there are sufficient active log data sets. If necessary, additional log data sets can be added dynamically using the **DEFINE LOG** command.

If the delay was caused by the lack of a resource required for offload, the necessary resource must be made available to allow offload to complete and thus permit logging to proceed. For information about recovery from this condition, see Archive log problems.

If the offload process has stalled because some resource is not available or for some other reason, it might be possible to resolve the problem by canceling the currently running offload task using the **ARCHIVE LOG CANCEL OFFLOAD** command. If there are hardware problems, it may be necessary to use z/OS commands to cancel the devices with problems.

If the problem occurred because archiving was set off, or because archive data sets could not be allocated, or for any other reason that requires the system parameters to be changed, the queue manager must be canceled as neither **STOP MODE(QUIESCE)** nor **STOP MODE(FORCE)** commands will work.

To free any tasks that are waiting because they were performing IBM MQ API calls that require logging, you must solve the underlying problem, or cancel the queue manager.

CSQJ112E
csect-name INSUFFICIENT ACTIVE LOG DATA SETS DEFINED IN BSDS

Explanation

There are not enough active log data sets defined in the BSDS to start the queue manager. This condition usually exists for one of the following reasons:

- Fewer than two data sets are defined for one of the active log copy sets.
- The CSQ6LOGP system parameters specified **TWOACTV=YES** but data sets for two copies of active log are not defined in BSDS.
- Fewer than two data sets are available (not flagged as STOPPED) for one of the active log copy sets.

System action

Startup is terminated.

System programmer response

Use the change log inventory utility to make the number of active log data sets defined in the BSDS consistent with the system parameters specified in CSQ6LOGP, or to add further active log data sets so that there are two or more active log data sets available for use in each active log copy. Restart the queue manager.

Note: Log data sets that are flagged as STOPPED will not be reused by IBM MQ. Once the queue manager has been restarted you might need to recover STOPPED log data sets. To clear the STOPPED status:

1. Stop the queue manager

2. Recover the log data set (either redefined or recovered from the other copy of the log)
3. Delete and re-add to the BSDS (using the change log inventory utility) with the appropriate RBAs

CSQJ113E
RBA *log-rba* NOT IN ANY ACTIVE OR ARCHIVE LOG DATA SET, CONNECTION-ID=*xxxx* THREAD-XREF=*yyyyyy*

Explanation

There was a request to read the log record starting at this RBA. However, this log record cannot be found in any active or archive log data set. The connection ID and thread-xref identify the user or application that encountered the problem (this could be an internal IBM MQ task). See Active log problems for information about dealing with problems on the log.

System action

Depending upon what log record is being read and why, the requestor might end abnormally with a reason code of X'00D1032A'.

System programmer response

Probable user error. Most likely, the archive log data set that contained the requested RBA has been deleted from the BSDS by the change log inventory utility. Locate the output from an old print log map run, and identify the data set that contains the missing RBA. If the data set has not been reused, run the change log inventory utility to add this data set back into the inventory of log data sets. Restart the queue manager.

CSQJ114I
ERROR ON ARCHIVE DATA SET, OFFLOAD CONTINUING WITH ONLY ONE ARCHIVE DATA SET BEING GENERATED

Explanation

An error occurred while accessing one of the archive data sets being created by offload. Because the dual archive option is specified, offload is continuing with the other archive data set. For the RBA range being offloaded, there is only one copy of archive instead of the usual two copies.

System action

Offload produces a single archive data set.

System programmer response

A second copy of this archive log data set can be made, and the BSDSs can be updated with the change log inventory utility.

CSQJ115E
OFFLOAD FAILED, COULD NOT ALLOCATE AN ARCHIVE DATA SET

Explanation

Offload could not allocate an archive log data set. The offload was not performed. This message is preceded by message CSQJ103E or CSQJ073E.

Note: If you are using the dual archiving option, neither copy is made.

System action

Offload will be tried at a later time.

System programmer response

Review the error status information of message CSQJ103E or CSQJ073E. Correct the condition that caused the data set allocation error so that, on retry, the offload can take place.

CSQJ116E
ERROR ADDING ARCHIVE ENTRY TO BSDS

Explanation

Offload could not add an archive entry to the BSDS. The offload is considered incomplete. The active log data set is not marked as reusable for new log data. This message is preceded by message CSQJ107E, CSQJ108E, or CSQJ109E.

System action

Offload will be retried at a later time.

System programmer response

See the specific preceding message for action.

CSQJ117E
INITIALIZATION ERROR READING BSDS DSNAME= *dsname*, ERROR STATUS=*eee*

Explanation

An error occurred during initialization reading from the specified BSDS. Error Status contains the VSAM return and feedback codes. It is a 2-byte field with the first containing the hexadecimal return code and the second byte containing the hexadecimal feedback code. See the *z/OS DFSMS Macro Instructions for Data Sets* manual for a description of VSAM return and reason codes.

System action

Startup is terminated.

System programmer response

Determine the cause of the read error using the VSAM error status information provided. Restart the queue manager.

CSQJ118E
MACRO *xxx* FAILED IN LOG INITIALIZATION, RC=*ccc*

Explanation

Log initialization received a return code from the named macro.

System action

Startup is terminated.

System programmer response

Determine the problem from the documentation on the named macro and return code. Then take appropriate steps, and restart the queue manager. If you do not have information about the named macro, note the macro name and the return code and contact IBM Support for help.

CSQJ119E
BOOTSTRAP ACCESS INITIALIZATION PROCESSING FAILED

Explanation

During queue manager initialization, the BSDS access function was unable to complete its initialization process. See Active log problems for information about dealing with problems on the BSDS or the log.

System action

Startup is terminated.

System programmer response

One or more error messages describing the specific error have preceded this message. See the specific messages for error analysis and the appropriate action to take.

CSQJ120E
DUAL BSDS DATA SETS HAVE UNEQUAL TIME STAMPS, SYSTEM BSDS1=*sys-bsds1*, BSDS2=*sys-bsds2*, UTILITY BSDS1=*uty-bsds1*, BSDS2=*uty-bsds2*

Explanation

When the queue manager was initialized, the time stamps of the dual BSDS did not agree. The time stamps from the system and from the change log inventory utility are shown for each BSDS. The time stamps have the format *date hh:mm:ss.th*.

System action

The queue manager attempts to re-synchronize the BSDS data sets to restore dual BSDS mode. If re-synchronization is successful, message CSQJ130I is issued and startup continues. Otherwise, startup is terminated.

System programmer response

If startup fails, run the print log map utility against each BSDS. From the output, determine which data set is obsolete, delete it, define a replacement for it, and copy the remaining BSDS to the replacement.

If output from the print log map utility for both data sets is similar, delete the data set with the oldest time stamp, and copy the data set with the most recent time stamp.

CSQJ121E
INITIALIZATION ERROR READING JFCB, DDNAME=*ddd*

Explanation

During queue manager initialization (if dual BSDS data sets are specified), the job file control block (JFCB) in z/OS is read to obtain the data set names associated with DDnames BSDS1 and BSDS2. This error is caused by a missing DD statement.

System action

Startup is terminated.

System programmer response

Ensure that a DD statement exists in the queue manager started task JCL procedure xxxxMSTR for DDname BSDS1. If dual BSDS data sets are used, ensure that a DD statement also exists in the queue manager started task JCL procedure xxxxMSTR for DDname BSDS2.

CSQJ122E
DUAL BSDS DATA SETS ARE OUT OF SYNCHRONIZATION

Explanation

During queue manager initialization, or when running a utility, the dual BSDSs were found to differ in content.

System action

The program or queue manager startup is terminated.

System programmer response

If the error occurred during queue manager initialization, run the print log map utility against each BSDS to determine which data set was last used as the first copy. Delete the second copy data set, define a replacement for the deleted data set, and copy the remaining BSDS to the replacement.

If the error occurred when running the BSDS conversion utility after the queue manager terminated abnormally, first attempt to restart the queue manager and shut it down cleanly before attempting to run the BSDS conversion utility again. If this does not solve the problem, run the print log map utility against each BSDS to determine which data set was last used as the first copy. Change the JCL used to invoke the BSDS conversion utility to specify this BSDS in the SYSUT1 DD statement, and remove the SYSUT2 DD statement, before submitting the job again.

CSQJ123E
CHANGE LOG INVENTORY FAILURE DETECTED

Explanation

During queue manager initialization, the BSDSs was found to have been incompletely processed by the change log inventory utility.

System action

Startup is terminated.

System programmer response

Run the print log map utility to determine what operation against the BSDS did not complete. Run the change log inventory utility against the BSDSs to allow any unfinished processing to be completed.

CSQJ124E
OFFLOAD OF ACTIVE LOG SUSPENDED FROM RBA *xxxxxx* TO RBA *xxxxxx* DUE TO I/O ERROR

Explanation

During offload, an unrecoverable input/output error was encountered on an active log data set. The data set experiencing the error is marked unusable, and no further logging is done to that data set.

System action

Active log data sets continue to be offloaded as they become full.

System programmer response

Recover the data manually from the data set, copy it to an archive data set, run the change log inventory utility to make the new archive data set available to the queue manager, and remove the error-prone active log data set.

CSQJ125E
ERROR COPYING BSDS, OFFLOAD CONTINUING WITHOUT THE BSDS COPY

Explanation

An error occurred while copying the BSDS data set during the offload process. The data set is not produced, and the volume containing the offloaded data set does not contain a BSDS for recovery use.

System action

The queue manager continues the offload process without producing a copy of the BSDS.

System programmer response

When archiving occurs, both a copy of the active log data set, and the BSDS at that time, are dumped. The BSDS is not critical because it will be copied again with the next archive log (the missing one simply means an extended restart time). However, the underlying data management problem (for example, not enough space allocated) should be resolved for subsequent BSDS offloads to occur.

CSQJ126E
BSDS ERROR FORCED SINGLE BSDS MODE

Explanation

An input/output error or a VSAM logical error occurred on a BSDS. This message is preceded by message CSQJ107E or CSQJ108E.

System action

IBM MQ enters single BSDS mode using the remaining BSDS.

System programmer response

Run an offline Access Method Services job to rename the error BSDS and define a new BSDS with the same name. Then enter the **RECOVER BSDS** command to reestablish dual BSDS mode.

CSQJ127I
SYSTEM TIME STAMP FOR BSDS=*date time*

Explanation

When the queue manager is initialized, the system time stamp for the BSDS is displayed. The time stamp is of the format *date hh:mm:ss.th*. This time stamp should be close to the last time at which this queue manager was stopped. If not, it might indicate a restart is being attempted with the wrong BSDS.

The time stamp will show as '****' if the BSDS has not been used before.

System action

Startup continues.

System programmer response

If the time displayed is not close to the time this queue manager was last stopped, and you cannot explain any time discrepancy, cancel the queue manager. From the queue manager started task JCL procedure xxxxMSTR, determine the data set names of the BSDSs and run the print log map utility. Check whether the active and archive log data sets all belong to this queue manager. If not, then change the started task JCL procedure xxxxMSTR for the queue manager to use the correct BSDSs.

CSQJ128E
LOG OFFLOAD TASK FAILED FOR ACTIVE LOG *dsname*

Explanation

The offload task ended abnormally while attempting to offload the RBA range in active log data set *dsname*.

System action

The offload task terminates and the archive data sets allocated to the offload task are deallocated and deleted. The status of the active log data sets involved in the unsuccessful offload processing remains set to 'not reusable'.

The log offload task will be re-initiated by one of several events. The most common are:

- All the available space in the current active log data set has been used (normal case)
- A CSQJ110E message is issued
- The queue manager address space is started, but data in the active log has not been archived
- An I/O error occurs on the active log, which will force the queue manager to truncate and offload the active log data set, and switch to a new active log data set

System programmer response

This message is the result of an offload error, and will be preceded by one or more IBM MQ messages (for example, CSQJ073E) and z/OS messages (for example, IEC030I, IEC031I, IEC032I). If the queue manager is operating with restricted active log resources (see message CSQJ110E), quiesce the system to restrict logging activity until the abnormal termination or the CSQJ110E condition can be resolved.

Investigate and correct the cause of the abnormal termination before the offload is attempted again by the queue manager.

CSQJ129E
END OF LOG RBA *eol-rba* COULD NOT BE FOUND IN ANY ACTIVE LOG DATA SET, HIGHEST RBA FOUND WAS *hi-rba*

Explanation

There was a request to find *eol-rba*, the log record that has been recorded in the BSDS as the highest RBA written. This RBA cannot be found in any active log data set. The highest RBA which could be found in any active data set was *hi-rba*.

System action

Startup processing is terminated.

System programmer response

Most likely, the active log data set containing the requested RBA has been deleted from the BSDS by the change log inventory utility. If the data set has not been reused, run the change log inventory utility to add this data set back into the BSDS. Restart the queue manager.

If the data set is not available, contact IBM Support.

CSQJ130I
DUAL BSDS MODE RESTORED FROM BSDS*n*

Explanation

Dual BSDS mode was restored using BSDS copy *n*. This is the BSDS data set with the most recent system time stamp.

System action

Startup continues.

CSQJ131E
csect-name ERROR WRITING QUEUE MANAGER INFORMATION TO Db2

Explanation

During command processing, a failure occurred attempting to write queue manager information to Db2.

System action

Processing of the command is terminated.

System programmer response

Check the console for messages relating to the problem.

CSQJ132E
csect-name ERROR READING QUEUE MANAGER INFORMATION FROM Db2

Explanation

During command processing, a failure occurred attempting to read queue manager information from Db2.

System action

Processing of the command is terminated.

System programmer response

Check the console for messages relating to the problem.

CSQJ133E
LRSN *rrr* NOT IN ANY ACTIVE OR ARCHIVE LOG DATA SET, CONNECTION-ID=*xxxx* THREAD-XREF=*yyyyyy*, QMGR=*qmgr-name*

Explanation

There was a request to read the log record starting at this LRSN for the indicated queue manager (which might not be the issuer of the message). However, this log record cannot be found in any active or archive log data set. The connection ID and thread-xref identify the user or application that encountered the problem (this could be an internal IBM MQ task). See Active log problems for information about dealing with problems on the log.

System action

Depending upon what log record is being read and why, the requestor might end abnormally with a reason code of X'00D1032A'.

System programmer response

This is probably a user error. Most likely, the archive log data set that contained the requested RBA has been deleted from the BSDS by the change log inventory utility. Locate the output from an old print log map run, and identify the data set that contains the missing LRSN. If the data set has not been reused, run the change log inventory utility to add this data set back into the inventory of log data sets. Restart the queue manager.

CSQJ134E
RBA *log-rba* NOT IN ANY ACTIVE OR ARCHIVE LOG DATA SET, CONNECTION-ID=*xxxx* THREAD-XREF=*yyyyyy*, QMGR=*qmgr-name*

Explanation

There was a request to read the log record starting at this RBA for the indicated queue manager. However, this log record cannot be found in any active or archive log data set. The connection ID and thread-xref identify the user or application that encountered the problem (this could be an internal IBM MQ task). See Active log problems for information about dealing with problems on the log.

System action

Depending upon what log record is being read and why, the requestor might end abnormally with a reason code of X'00D1032A'.

System programmer response

This problem can occur for the following reasons:

1. The entry with the log range in the BSDS has been deleted from the BSDS
2. The entry with the log range is in BSDS, but the archive log data set has been deleted. When an archive log is created, the CSQ6ARVP parameter **ARCRETN** is used to specify when the data set can be deleted. When this date has passed MVS deletes the data set, so if you are trying to use this data set after this date, the data set cannot be found.

See BSDS problems for further information.

CSQJ136I
UNABLE TO ALLOCATE TAPE UNIT FOR CONNECTION-ID=*xxxx* CORRELATION-ID= *yyyyyy*, *m* ALLOCATED *n* ALLOWED

Explanation

An attempt to allocate a tape unit for the indicated connection ID failed. The current maximum tape unit specified is *n*, but only *m* are physically available.

System action

The process for the connection ID and correlation ID is held until either an allocated tape unit becomes free or more tape units are varied online and made available to the archive read task. This situation rectifies itself over time as currently allocated tape units become available.

CSQJ139I
LOG OFFLOAD TASK ENDED

Explanation

Processing of the active log offload ended.

System action

This message is written to the z/OS console.

CSQJ140I
Data set *dsname* successfully added to active log copy *n*

Explanation

A DEFINE LOG command has dynamically added a new log data set, *dsn*, and added it to either the LOGCOPY1 or LOGCOPY2 ring of active log data sets, as indicated by *n*.

The new active log data set is eligible to be used when the current active log data set fills and logging switches to the next active log data set in the ring.

Information about the data set is stored in the BSDS and will persist over a restart of the queue manager.

CSQJ141E
Error adding new active log data set *dsname*

Explanation

A DEFINE LOG command failed to add a new log data set. Further information about the failure is given in the preceding messages.

System programmer response

Investigate and correct the cause of the failure, then enter the command again.

CSQJ142I
Data set *dsname* has been used previously

Explanation

IBM MQ checks that a data set being added by a **DEFINE LOG** command has not been previously used as a log data set, as this might be an indication of operator error. The requested data set *dsname* was found to have been previously so used.

System action

The data set is closed and deallocated. Dynamic addition of a new active log data set fails.

System programmer response

Ensure that the data set being added as an active log data set is newly allocated, or has been formatted with the active log preformat utility, CSQJUFMT.

CSQJ143I
BSDS active log data set record is full

Explanation

The maximum number of active log data sets is fixed. No further entries can be inserted in the BSDS after the maximum has been reached.

System action

Dynamic addition of a new active log data set fails.

CSQJ144I
Active log data set allocation error

Explanation

It was not possible for IBM MQ to dynamically allocate the requested data set (named in the following CSQJ141E message) for use as a new active log data set.

System action

Dynamic addition of a new active log data set fails.

System programmer response

Ensure that the data set being added as a new active log data set is a VSAM linear data set with SHAREOPTIONS(2 3) and that it is not in use by any other jobs.

CSQJ150E
LOG CAPTURE EXIT ABEND, EXIT DEACTIVATED

Explanation

An abnormal program interrupt was detected while executing in the installation-supplied log capture exit code (that is entry point CSQJW117 in load module CSQJL004). As a result of this, the log capture exit will no longer be active; log data will no longer be available for exit capture/processing.

This message can only occur when an *installation-supplied* log capture exit (entry CSQJW117) is active for this queue manager.

System action

The log capture exit (entry point CSQJW117) is terminated. No further calls will be attempted for this queue manager. A full dump is provided for diagnostic purposes.

System programmer response

Determine the cause of the CSQJL004 load module (CSQJW117 entry point) abend and take corrective action.

Note: A correctly-functioning copy of load module CSQJL004/entry CSQJW117 must be available to start the queue manager. If the problem that caused this error cannot be corrected, ensure that the default CSQJW117 entry (load module CSQJL004 - supplied with IBM MQ) is available during the next queue manager start.

CSQJ151I
csect-name ERROR READING RBA *rrr*, CONNECTION-ID=*xxxx* CORRELATION-ID=*yyyyyy* REASON CODE=*ccc*

Explanation

The queue manager could not successfully complete the read of the indicated RBA due to reason code *ccc*. The user or application that encountered the error is identified by the connection and correlation IDs. Messages that have the same connection ID and correlation ID relate to the same application. Correlation IDs beginning with '0nn', where nn is a number from 01 to 28, identify system agents.

System action

The queue manager attempts to recover from the error.

System programmer response

If the queue manager was able to recover from the error and successfully complete the application, no further action is required. If the application abnormally terminated or the queue manager could not recover successfully, this message is followed by one or more messages. Refer to the information in this message and the subsequent messages to determine the appropriate corrective action. .

CSQJ152I
csect-name ERROR BUILDING ARCHIVE LOG VOLUME REPORT, CONNECTION-ID=*xxxx* CORRELATION-ID=*yyyyyy* REASON CODE=*ccc*

Explanation

An error occurred while attempting to create the archive log volume report. An RBA range could not be successfully mapped into one or more archive data sets due to reason code *ccc*. The user or application that encountered the error is identified by the connection and correlation IDs. This message might be preceded by one or more related error messages. Messages that have the same connection ID and correlation ID relate to the same application. Correlation IDs beginning with '0nn', where nn is a number from 01 to 28, identify system agents.

This failure could be caused by one or more missing archive log data sets, or a system error (for example, an I/O error reading the BSDS).

System action

The archive log volume report (see message CSQJ330I) is not produced. In addition, no premounting of tapes is possible.

The user or application continues processing. The physical read process for the user or application continues until the job completes normally or terminates abnormally. The job can terminate abnormally if the error is encountered again when the data set is physically required for the read process.

System programmer response

If the user or application completes successfully, no further action is necessary. If the user or application does not complete successfully, refer to the messages related to the actual failure to determine the appropriate corrective action.

CSQJ153I
csect-name ERROR READING LRSN *rrr*, CONNECTION-ID=*xxxx* CORRELATION-ID=*yyyyyy* REASON CODE=*ccc*, QMGR=*qmgr-name*

Explanation

The queue manager could not successfully complete the read of the indicated LRSN for the indicated queue manager (which might not be the issuer of the message) due to reason code *ccc*. The user or application that encountered the error is identified by the connection and correlation IDs. Messages that have the same connection ID and correlation ID relate to the same application. Correlation IDs beginning with '0*nn*', where *nn* is a number from 01 to 28, identify system agents.

System action

The queue manager attempts to recover from the error.

System programmer response

If the queue manager was able to recover from the error and successfully complete the application, no further action is required. If the application abnormally terminated or the queue manager could not recover successfully, this message is followed by one or more messages. Refer to the information in this message and the subsequent messages to determine the appropriate corrective action..

CSQJ154I
csect-name ERROR READING RBA *rrr*, CONNECTION-ID=*xxxx* CORRELATION-ID=*yyyyyy* REASON CODE=*ccc*, QMGR=*qmgr-name*

Explanation

The queue manager could not successfully complete the read of the indicated RBA for the indicated queue manager due to reason code *ccc*. The user or application that encountered the error is identified by the connection ID and correlation ID. Messages that have the same connection ID and correlation ID relate to the same application. Correlation IDs beginning with '0nn', where nn is a number from 01 to 28, identify system agents.

System action

The queue manager attempts to recover from the error.

System programmer response

If the queue manager was able to recover from the error and successfully complete the application, no further action is required. If the application abnormally terminated or the queue manager could not recover successfully, this message is followed by one or more messages. Refer to the information in this message and the subsequent messages to determine the appropriate corrective action.

CSQJ155E
csect-name ACTIVE LOG DSNAME=*xxxxx* IS LARGER THAN 4GB

Explanation

The queue manager has opened a LOG dataset that has space allocated that is larger than 4GB. The maximum LOG size supported is 4GB.

System action

Processing continues. The LOG dataset continues to be used but the space used is smaller than the space allocated. The allocated highest RBA value overflows a 32bit word and it is truncated. The truncated highest RBA is used resulting in a smaller LOG dataset.

System programmer response

The LOG dataset should be allocated with space not exceeding 4GB. Use Access Method Services to define the data set with a maximum size of 4 GB. Refer to Log dataset definitions to determine the LOG dataset size.

To reallocate LOG datasets with space smaller or equal to 4GB, carry out these steps:

1. Allocate and format new COPY1, and COPY2 if used, active log datasets with a maximum size of 4GB, and run CSQJUFMT if required.
2. Dynamically add the new logs to the queue manager with command **DEFINE LOG**.
3. Use command **ARCHIVE LOG** to roll into the new logs.
4. Shut down the queue-manager when one of the new logs is the current active log dataset.
5. Use the CSQJU003 utility to remove the active logs which are too small. First run CSQJU004 to display the log status in BSDS to make sure the ones to be removed are 'REUSABLE', that is they have been successfully archived.
6. Restart the queue manager. There should not be any CSQJ115E messages issued.

CSQJ160I
LONG-RUNNING UOW FOUND, URID=*urid* CONNECTION NAME=*name*

Explanation

During log switch processing an uncommitted unit of recovery, spanning more than two active log switches, has been encountered. The unit of recovery identifier *urid* together with the connection name *name* identify the associated thread.

System action

Processing continues.

System programmer response

Consult with the application programmer to determine if there is a problem with the unit of recovery, and to ensure that the application commits work frequently enough. Uncommitted units of recovery can lead to difficulties later.

If required, issue the command `DISPLAY CONN(*) WHERE(QMURID EQ urid) ALL` to display more information about the unit of recovery, including the channel name for remote connections.

CSQJ161I
UOW UNRESOLVED AFTER *n* OFFLOADS, URID=*urid* CONNECTION NAME=*name*

Explanation

During log switch processing, an uncommitted unit of recovery was encountered that now has activity spanning several log data sets. The unit of recovery identifier *urid* together with the connection name *name* identify the associated thread.

System action

Processing continues.

System programmer response

Consult with the application programmer to determine if there is a problem with the unit of recovery, and to ensure that the application commits work frequently enough. Uncommitted units of recovery can lead to difficulties later.

If required, issue the command `DISPLAY CONN(*) WHERE(QMURID EQ urid) ALL` to display more information about the unit of recovery, including the channel name for remote connections.

CSQJ163E
COPY(2) specified but TWOACTV(NO)

Explanation

A **DEFINE LOG** command specified the **COPY(2)** parameter but the dual logging parameter (**TWOACTV=YES**) was not specified in CSQ6LOGP at queue manager startup.

System action

Dynamic addition of the new active log data set fails.

System programmer response

Either specify **COPY(1)** on the **DEFINE LOG** command or configure the queue manager to use dual logging.

CSQJ164I
csect-name Log archiving delayed, all available offload tasks in use

Explanation

The offload of one or more active logs has been delayed because all available offload tasks are in use.

A maximum of 31 offload tasks can concurrently write new archive log data sets. The number of offload tasks can be tuned using the **MAXCNOFF** parameter, which is set using either CSQ6LOGP or the SET LOG command. **MAXCNOFF** is provided to tailor the offloading of IBM MQ logs to match system constraints, such as the number of available tape units.

System action

Processing continues. The offload will complete when an offload task becomes available. Message CSQJ168I will be issued when the offload of active logs is no longer being delayed.

System programmer response

This is most likely a transient situation as a result of IBM MQ suddenly being able to archive a large number of full active logs, for example after problems with archiving have been resolved.

In other circumstances, review the **MAXCNOFF** parameter setting.

Consider increasing the active log capacity to match the active and archive log rates. The DEFINE LOG command can be used to provide additional active log capacity.

CSQJ166E

zHyperWrite configuration is inconsistent for active log copy *n*

Explanation

The data sets for each copy of the active log should be consistently configured so that they are capable of being used with zHyperWrite.

This means that, either all of the data sets that comprise an active log copy should be on zHyperWrite capable volumes, or none of the data sets should be on zHyperWrite capable volumes.

The queue manager has detected an inconsistency in the zHyperWrite configuration for the data sets that comprise log copy *n*.

System action

Processing continues, but an inconsistent logging rate might be observed due to the inconsistent configuration of the active log volumes.

System programmer response

Review the configuration for the active log volumes.

The status of zHyperWrite log capability is asserted at queue manager start up and therefore might be stale if the capability changes.

CSQJ167E

ZHYWRITE(YES) specified but no active logs are zHyperWrite capable

Explanation

Prior to IBM MQ 9.4.0, the system parameter **ZHYWRITE** has been set to YES, but the queue manager cannot exploit zHyperWrite because none of the active log copies are on zHyperWrite capable volumes.

From IBM MQ 9.4.0, the system parameter **ZHYWRITE** has been set to YES, but the queue manager detects that none of the active log copies are on zHyperWrite capable volumes. The status of zHyperWrite log capability is asserted at queue manager start up, and therefore might be stale if the capability changes. The queue manager will issue log writes with zHyperWrite enabled. regardless of whether the active log copies are on zHyperWrite capable volumes. See Using zHyperWrite with IBM MQ active logs for more information.

From IBM MQ 9.4.0, this message might be issued because **ZHYWRITE** was enabled, due to **ZHYLINK** being set on the queue manager.

System action

Prior to IBM MQ 9.4.0, processing continues without log writes being enabled for zHyperWrite. From IBM MQ 9.4.0, processing continues with log writes being enabled for zHyperWrite.

System programmer response

Review the configuration for the active log volumes and the zHyperWrite feature.

From IBM MQ 9.4.0, IBM MQ will still attempt to use zHyperWrite regardless of this message being output. It can sometimes take a while for zHyperWrite to be enabled at the z/OS and disk level following an IPL. Once zHyperWrite is enabled, IBM MQ log writes will make use of zHyperWrite.

CSQJ168I

csect-name Log archiving is no longer delayed

Explanation

The offload of active logs is no longer being delayed by a shortage of available offload tasks.

System action

Processing continues.

CSQJ169E

LAST CHECKPOINT NOT FOUND IN ACTIVE LOG COPY AND WITH STARTRBA=*&*, CHECKPOINT RBA=*&*.

Explanation

During active log switch processing the last checkpoint was not found on any active logs. This could leave the queue manager in an unrecoverable position if there are insufficient archive logs available to find the required recovery point during restart processing.

This can be an indication that checkpoint processing might have stalled, or is not completing in a timely manner, and should be investigated.

System action

Log switch processing continues.

System programmer response

You might be able to re-establish checkpointing by stopping and restarting the queue manager. If checkpointing is stalled, the STOP QMGR command might not be able to shut down the queue manager normally. If this happens, you might need to cancel the queue manager. Before doing so, ensure that the logs from the restart RBA onwards are available. You can find the restart RBA using the DISPLAY USAGE command.

If it appears that checkpointing has stalled, take a dump of the queue manager address space, and contact IBM Support for assistance to help understand why checkpointing might have stalled.

If checkpointing does not appear to have stalled, an alternative reason for this situation might be that the active logs for the queue managers are too small for the current workload, and checkpoint processing is not completing during the scope of the lifespan of one active log.

CSQJ200I
csect-name UTILITY PROCESSING COMPLETED SUCCESSFULLY

Explanation

The utility completed successfully.

CSQJ201I
csect-name UTILITY PROCESSING WAS UNSUCCESSFUL

Explanation

The utility was unable to complete processing successfully.

System action

The current utility is terminated.

System programmer response

Review other messages produced by the utility to determine the appropriate action to be taken.

CSQJ202E
INSUFFICIENT STORAGE AVAILABLE TO CONTINUE

Explanation

A request for storage was unsuccessful because no more storage is available.

System action

The current utility is terminated.

System programmer response

Rerun the utility after increasing the storage available.

CSQJ203E
oper OPERATION IS INVALID

Explanation

The user entered a utility control statement operation (*oper*) that is invalid.

System action

The current utility is terminated.

System programmer response

Correct the control statement, and rerun the utility.

CSQJ204E
xxxx PARAMETER IS INVALID

Explanation

The user specified a utility control statement parameter (*xxxx*) that is invalid.

System action

The current utility is terminated.

System programmer response

Correct the control statement, and rerun the utility.

CSQJ205E
xxxx PARAMETER HAS NO ARGUMENT

Explanation

xxxx contains the name of a parameter that requires an argument.

System action

The current utility is terminated.

System programmer response

Specify an argument for the identified parameter and then rerun the utility.

CSQJ206E
xxxx PARAMETER REQUIRES NO ARGUMENT

Explanation

xxxx contains the name of the parameter that has been incorrectly followed by an = symbol.

System action

The current utility is terminated.

System programmer response

Correct the control statement, and rerun the utility.

CSQJ207E
PARAMETERS INCONSISTENT WITH SPECIFIED OPERATION

Explanation

The user has specified utility control statement parameters that are inconsistent with the specified utility operation.

System action

The current utility is terminated.

System programmer response

Correct the control statement, and rerun the utility.

CSQJ211E
UNEXPECTED END OF DATA ON SYSIN DATA SET

Explanation

Additional control statements were expected, but could not be found.

System action

The current utility is terminated.

System programmer response

Correct the control statements, and rerun the utility.

CSQJ212E
ERROR RETURNED FROM BSDS READ, RPLERRCD= *yy*, DDNAME=*ddd*

Explanation

A VSAM GET was issued that resulted in a nonzero return code. *yy* contains the error code returned by VSAM. *ddd* contains the DDname of the BSDS encountering the error.

System action

The current utility is terminated.

System programmer response

The action taken is dictated by the reason code. See RPLERRCD for information about the reason code in RPLERRCD. The BSDS might have to be recovered by use of a backup copy.

CSQJ213E
ERROR RETURNED FROM BSDS WRITE, RPLERRCD= *yy*, DDNAME=*ddd*

Explanation

A VSAM PUT was issued that resulted in a nonzero return code. *yy* contains the error code returned by VSAM. *ddd* contains the DDname of the BSDS encountering the error.

System action

The current utility is terminated.

System programmer response

The action to be taken is dictated by the reason code. See RPLERRCD for information about the reason code in RPLERRCD. The BSDS might have to be recovered by use of a backup copy.

If this error occurs when running the BSDS conversion utility (CSQJUCNV), and RPLERRCD indicates that the reason was an attempt to store a record with a duplicate key, ensure that the output BSDS is empty before running the utility.

CSQJ214E
SPECIFIED DSNAME ALREADY EXISTS IN BSDS, DDNAME=*ddd*

Explanation

You attempted a NEWLOG operation with a data set name that already exists in the BSDS. An entry is never made in a BSDS if the specified DSNAME currently exists in either the active or archive records of that BSDS. *ddd* contains the DDname of the subject BSDS.

System action

The current utility is terminated.

System programmer response

Either correct the control statement and rerun the utility, or delete the existing DSNAME from the BSDS and rerun the utility.

CSQJ215I
modname timestamps formatted with no local correction

Explanation

The parameter **TIME(RAW)** was specified on the invocation of utility *modname*. Where possible, timestamps formatted as date and time in the output will have no local timezone, or leapsecond adjustment performed so will be the UTC time of the event on the source system.

This mode of processing is most useful when the log, or BSDS being formatted has been produced on a remote system in a different timezone, or in a different daylight saving regime.

System action

Processing continues.

System programmer response

Either correct the control statement and rerun the utility, or delete the existing DSNAME from the BSDS and rerun the utility.

CSQJ216E
BSDS ACTIVE LOG DATA SET RECORD IS FULL, DDNAME=*ddd*

Explanation

The maximum number of active log data sets is fixed. No further entries can be inserted in the BSDS after the maximum has been reached. *ddd* contains the DDname of the subject BSDS.

System action

The current utility is terminated.

System programmer response

Run the print log map utility to determine the current status of the BSDS. Subsequent actions can then be formulated, depending upon the status of the BSDS.

CSQJ217E
SPECIFIED DSNAME DOES NOT EXIST IN BSDS, DDNAME=*ddd*

Explanation

The **DELETE** operation specifies a DSNAME that cannot be found in the BSDS. *ddd* contains the DDname of the subject BSDS.

System action

The current utility is terminated.

System programmer response

Correct the control statement, and rerun the utility.

CSQJ218E
SPECIFIED VOLUME DOES NOT EXIST IN BSDS, DDNAME=*ddd*

Explanation

The **DELETE** operation specifies a **COPY1VOL** or **COPY2VOL** argument that cannot be found in the BSDS. *ddd* contains the DDname of the subject BSDS.

System action

The current utility is terminated.

System programmer response

Correct the control statement, and rerun the utility.

CSQJ219E
OPEN ERROR, DDNAME=*ddd*

Explanation

An error occurred when *csect-name* tried to open a data set named *ddd*.

This error can be caused by a number of different conditions. The most probable conditions are:

1. The DDname of the SYSPRINT, SYSIN, or SYSUT1 data set was not specified in the user's job control language (JCL)
2. The queue manager is currently active
3. The BSDS has been allocated by another job with a disposition (DISP) that conflicts with the DISP specified in the user's JCL
4. The data set associated with *ddd* is already open, possibly due to an earlier error
5. The user is not authorized to access the data set associated with *ddd*
6. Insufficient storage is available to perform the OPEN operation
7. The catalog indicates that the data set associated with *ddd* has an invalid physical record size

System action

The current utility is terminated.

System programmer response

The user's action depends on the condition that caused the OPEN error. The following is a list of appropriate actions corresponding to the conditions listed in the explanation:

1. Provide the missing data definition (DD) statements, and then rerun the utility. See the section Preparing your program to run for further information.
2. Wait until the queue manager is inactive before running the utility again because the log utility cannot run while it is active.
3. Correct the disposition conflict and then rerun the utility.
4. Submit an Access Method Services (IDCAMS) VERIFY job against the data set associated with *ddd*. Rerun the log utility job.
5. In the case of an authorization problem, a separate message is usually generated from the authorization facility (RACF, for example). Investigate the authorization messages and obtain the proper authorization before running the utility again.
6. Insufficient storage is usually accompanied by a separate error from z/OS. Increase the storage available and rerun the utility.
7. Reallocate the data set with a suitable physical record size.

CSQJ220E
BSDS IN CREATE MODE. NO DATA TO MAP, DDNAME=*ddd*

Explanation

A utility found the BSDS to be in create mode, so it cannot contain data to map. *ddd* contains the DDname of the data set.

System action

The current utility is terminated.

System programmer response

Correct the JCL so that a non-null data set can be processed.

CSQJ221I
PREVIOUS ERROR CAUSED *oper* OPERATION TO BE BYPASSED

Explanation

Errors were encountered errors during utility processing. These errors subsequently caused *oper* to be bypassed.

This message is a warning only and is displayed after messages that specify the error or errors that occurred. Note that the error or errors might not be associated with the current *oper* operation; rather, under log utility processing, a significant error in any operation causes the control statements for this and any subsequent operations to be checked for syntax only. BSDS updates do not occur for any operation specified in this message.

System action

The log utility continues to process. However, for this and all subsequent operations, the BSDS is not updated and the utility only checks the syntax of the control statements.

System programmer response

Consult the previous messages and correct any errors that caused this message to be generated. Resubmit the log utility job for all operations that have been bypassed.

CSQJ222E
INVALID SPECIFICATION OF *xxxx* PARAMETER ARGUMENT

Explanation

You specified the parameter *xxxx*. This parameter is not valid for the argument.

System action

The current utility is terminated.

System programmer response

Correct the parameter argument on the control statement, and rerun the utility.

CSQJ223E
xxxx PARAMETER ARGUMENT EXCEEDS MAXIMUM ALLOWABLE LENGTH

Explanation

xxxx specifies the name of the parameter with an argument value that exceeded the maximum length allowed.

System action

The current utility is terminated.

System programmer response

Correct the parameter argument on the control statement, and rerun the utility.

CSQJ224E
xxxx PARAMETER APPEARS TOO OFTEN

Explanation

xxxx gives the name of the parameter that you have specified more than once on the same control statement.

System action

The current utility is terminated.

System programmer response

Remove the redundant parameter, and rerun the utility.

CSQJ225I
oper OPERATION SUCCESSFULLY COMPLETED

Explanation

The *oper* specified in the message identifies the name of the change log inventory utility operation that has been successfully completed.

CSQJ226E
SPECIFIED VOLUME ALREADY EXISTS IN BSDS, DDNAME=*ddd*

Explanation

The specified volume currently exists in the archive log records of the BSDS. *ddd* specifies the DDname of the subject BSDS.

System action

The current utility is terminated.

System programmer response

Either correct the parameter argument on the control statement, or delete the specified volume and rerun the utility.

CSQJ227E
NO SPACE IN BSDS FOR ADDITIONAL ARCHIVE ENTRIES, DDNAME=*ddd*

Explanation

The maximum number of archive volumes has been exceeded, and no more space is available for volume entries in the copy specified.

System action

The current utility is terminated.

System programmer response

Delete some of the archive entries in the specified copy number, and rerun the utility.

CSQJ228E
csect-name LOG DEALLOCATION ERROR DSNAME=*dsname*, ERROR STATUS=*eeeeiiii*, SMS REASON CODE=*ssssssss*

Explanation

An error occurred when trying to dynamically deallocate the data set. Error status is the error reason code returned by z/OS dynamic allocation.

System action

Processing continues.

System programmer response

The error status portion of this message contains a 2-byte error code (*eeee*, S99ERROR) followed by the 2-byte information code (*iiii*, S99INFO) from the SVC99 request block. If the S99ERROR code indicates an SMS allocation error ('97xx'), then *ssssssss* contains additional SMS reason code information obtained from S99ERSN.

See the Interpreting DYNALLOC return codes topic of the *z/OS MVS Authorized Assembler Services Guide* for information about these codes .

CSQJ230E
LOG OFFLOAD INITIALIZATION PROCESSING FAILED

Explanation

During queue manager initialization, the offload function was unable to complete its initialization process.

System action

Startup is terminated.

System programmer response

One or more error messages describing the specific error preceded this message. See the specific messages for error analysis and the appropriate actions to take.

CSQJ231E
LOG COMMAND INITIALIZATION PROCESSING FAILED

Explanation

During queue manager initialization, the command function was unable to complete its initialization process.

System action

Startup is terminated.

System programmer response

One or more error messages describing the specific error preceded this message. See the specific messages for error analysis and the appropriate action to take.

CSQJ232E
OUTPUT DATA SET CONTROL INITIALIZATION PROCESSING FAILED

Explanation

During queue manager initialization, the output data set control function was unable to complete its initialization process.

System action

Startup is terminated.

System programmer response

One or more error messages describing the specific error preceded this message. See the specific message for error analysis and the appropriate action to take.

CSQJ233E
ARCHIVE LOG READ INITIALIZATION PROCESSING FAILED

Explanation

During queue manager initialization, the archive log read function was unable to complete its initialization process.

System action

Startup is terminated.

System programmer response

One or more error messages describing the specific error preceded this message. See the specific messages for error analysis and the appropriate action to take.

CSQJ234E
ARCHIVE LOG COMMAND QUIESCE INITIALIZATION PROCESSING FAILED

Explanation

During queue manager initialization, the quiesce function which supports the **ARCHIVE LOG MODE(QUIESCE)** command processing was unable to complete its initialization process.

System action

Startup is terminated.

System programmer response

One or more error messages describing the specific error preceded this message. See the specific messages for error analysis and the appropriate action to take.

CSQJ235E
OUTPUT BUFFER WRITER INITIALIZATION PROCESSING FAILED

Explanation

During queue manager initialization, the output buffer writer function was unable to complete its initialization process.

System action

Startup is terminated.

System programmer response

One or more error messages describing the specific error preceded this message. See the specific messages for error analysis and the appropriate action to take.

CSQJ236E
BOOTSTRAP ACCESS TERMINATION PROCESSING FAILED

Explanation

During queue manager termination, the BSDS access function was unable to complete its termination process.

System action

Termination processing continues.

System programmer response

One or more error messages describing the specific error preceded this message. See the specific messages for error analysis and the appropriate action to take.

CSQJ238E
LOG OFFLOAD TERMINATION PROCESSING FAILED

Explanation

During queue manager termination, the offload function was unable to complete its termination process.

System action

Termination processing continues.

System programmer response

One or more error messages describing the specific error preceded this message. See the specific messages for error analysis and the appropriate action to take.

CSQJ239E
LOG COMMAND TERMINATION PROCESSING FAILED

Explanation

During queue manager termination, the command function was unable to complete its termination process.

System action

Termination processing continues.

System programmer response

One or more error messages describing the specific error preceded this message. See the specific messages for error analysis and the appropriate action to take.

CSQJ240E
OUTPUT DATA SET CONTROL TERMINATION PROCESSING FAILED

Explanation

During queue manager termination, the output data set control function was unable to complete its termination process.

System action

Termination processing continues.

System programmer response

One or more error messages describing the specific error preceded this message. See the specific messages for error analysis and the appropriate action to take.

CSQJ241E
ARCHIVE LOG READ TERMINATION PROCESSING FAILED

Explanation

During queue manager termination, the archive log read function was unable to complete its termination process.

System action

Termination processing continues.

System programmer response

One or more error messages describing the specific error preceded this message. See the specific messages for error analysis and the appropriate action to take.

CSQJ242E
ARCHIVE LOG COMMAND QUIESCE TERMINATION PROCESSING FAILED

Explanation

During queue manager termination, the quiesce function which supports the **ARCHIVE LOG MODE(QUIESCE)** command processing was unable to complete its termination process.

System action

Termination processing continues.

System programmer response

One or more error messages describing the specific error preceded this message. See the specific messages for error analysis and the appropriate action to take.

CSQJ243E
OUTPUT BUFFER WRITER TERMINATION PROCESSING FAILED

Explanation

During queue manager termination, the output buffer writer function was unable to complete its termination process.

System action

Termination processing continues.

System programmer response

One or more error messages describing the specific error preceded this message. See the specific messages for error analysis and the appropriate action to take.

CSQJ244E
MACRO *xxx* FAILED IN LOG TERMINATION, RC=*ccc*

Explanation

During termination, there was a return code from the named macro that indicated an error.

System action

Termination processing continues.

System programmer response

If the problem persists, contact IBM Support for assistance.

CSQJ245D
RESTART CONTROL INDICATES TRUNCATION AT RBA *rrr*. REPLY Y TO CONTINUE, N TO CANCEL

Explanation

The conditional restart control record in use indicates that the log should be truncated at the specified RBA.

System action

If 'Y', queue manager startup continues. If 'N', startup is terminated.

System programmer response

Run the change log inventory utility (CSQJU003) to modify the conditional restart record.

CSQJ246D
RESTART CONTROL INDICATES COLD START AT RBA *rrr*. REPLY Y TO CONTINUE, N TO CANCEL

Explanation

The conditional restart control record in use indicates that the queue manager is to be restarted and that logging is to begin at the specified RBA.

System action

If 'Y', queue manager startup continues. If 'N', startup is terminated.

System programmer response

Run the change log inventory utility (CSQJU003) to modify the conditional restart record.

CSQJ247E
csect-name I/O ERROR PROCESSING BSDS ARCHIVE LOG RECORD, RC=*rc* REASON=*reason*

Explanation

An input/output error occurred while processing a BSDS record. *rc* indicates the return code received from the input/output operation. *reason* indicates the reason code received from the operation.

Return code 4 indicates that IBM MQ detected a problem. Return code 8 indicates a VSAM error.

System action

Startup is terminated.

System programmer response

For a return code of 4, if the problem persists, contact IBM Support for assistance. For a return code of 8, run an offline Access Method Services job to determine the cause of the VSAM error.

CSQJ250I
csect-name DATA SET *dsname* HAS SHAREOPTIONS LESS THAN (2 3) - CF STRUCTURE RECOVERY NOT POSSIBLE

Explanation

An active log data set was detected with share options that do not permit CF structure recovery in a queue sharing group environment. All active log data sets must be have SHAREOPTIONS(2 3) at least to allow CF structure recovery.

This can occur when the queue manager's own log data sets are checked during startup, or when a **RECOVER CFSTRUCT** command is issued that requires to access another queue manager's log data sets.

System action

If this is a result of a **RECOVER CFSTRUCT** command, the command is terminated. Otherwise, startup continues, but CF structure recovery will not be possible.

System programmer response

If you want CF structure recovery, use the Access Method Services ALTER function to correct the SHAREOPTIONS for the data set; for example

```
ALTER dsname.DATA SHAREOPTIONS(2 3)
```

Then restart the queue manager that owns the data set.

CSQJ295D
RESTART CONTROL INDICATES TRUNCATION AT LRSN *rrr*. REPLY Y TO CONTINUE, N TO CANCEL

Explanation

The conditional restart control record in use indicates that the log should be truncated at the specified LRSN.

System action

If 'Y', queue manager startup continues. If 'N', startup is terminated.

System programmer response

Run the change log inventory utility (CSQJU003) to modify the conditional restart record.

CSQJ301E
csect-name ERROR USING ONLINE BOOTSTRAP DATA SET (ACTION CODE *a*)

Explanation

During command processing for the **RECOVER BSDS** command or the **ARCHIVE LOG** command, an error occurred while performing an operation on the BSDS. The type of operation is specified by the code *a*:

1
Unable to OPEN the BSDS

2
Unable to read a required record from the BSDS

3
Unable to write a required record to the BSDS

4
The contents of the stable BSDS was successfully copied to the replacement BSDS; however, the queue manager was unable to successfully restore dual BSDS operation

System action

If this message was received during processing of the **RECOVER BSDS** command, then the queue manager will continue in single BSDS mode. If this message was received during processing of the **ARCHIVE LOG** command, the archive log history record in the BSDS will not be updated to reflect the occurrence of an **ARCHIVE LOG** command; logging and the offload processing will continue.

System programmer response

If this message was received during processing of the **RECOVER BSDS** command, recovery action must be performed on the BSDS before re-entering the command. If this message was received during processing of the **ARCHIVE LOG** command, no action is necessary.

CSQJ302E
ALLOCATION ERROR ON REPLACEMENT BSDS DSNAME=*dsname* ERROR STATUS=*eee*

Explanation

The **RECOVER BSDS** command encountered an error while trying to allocate the specified data set dynamically. DSNAME is the data set name. Error Status is the error code and information code returned by z/OS dynamic allocation.

System action

Processing of the command is terminated. The queue manager continues in single BSDS mode.

System programmer response

Determine the cause of the error from the error status contained in the message, and correct the condition. Then re-enter the **RECOVER BSDS** command.

The error status portion of this message contains the 2-byte error code (S99ERROR) followed by the 2-byte information code (S99INFO) from the SVC request block.

See the Interpreting DYNALLOC return codes topic of the *z/OS MVS Authorized Assembler Services Guide* for information about these codes .

CSQJ303E
WRITE ERROR ON REPLACEMENT BSDS DSNAME=*dsname* ERROR STATUS=*eee*

Explanation

The **RECOVER BSDS** command encountered an error while attempting to write to the specified BSDS. Error status contains the VSAM return and feedback codes. It is a 2-byte field with the first containing the hexadecimal return code and the second containing the hexadecimal feedback code.

System action

Processing of the command is terminated. The queue manager continues in single BSDS mode.

System programmer response

Run an offline Access Method Services job to delete or rename the replacement BSDS and define a new BSDS with the same name. Re-enter the **RECOVER BSDS** command to reestablish dual BSDS mode.

CSQJ304E
ERROR CLOSING REPLACEMENT BSDS DSNAME=*dsname* ERROR STATUS=*eee*

Explanation

The **RECOVER BSDS** command encountered an error while attempting to close the specified BSDS. Error Status contains the VSAM return and feedback codes. It is a 2-byte field with the first containing the hexadecimal return code and the second containing the hexadecimal feedback code.

System action

Processing of the command is terminated. The queue manager continues in single BSDS mode.

System programmer response

Run an offline Access Method Services job to delete or rename the replacement BSDS and define a new BSDS with the same name. Re-enter the **RECOVER BSDS** command to reestablish dual BSDS mode.

CSQJ305E
REPLACEMENT BSDS NOT EMPTY DSNAME=*dsname*

Explanation

The **RECOVER BSDS** command was issued, but the replacement BSDS was not empty; that is, it contained data.

System action

Processing of the command is terminated. The queue manager continues in single BSDS mode.

System programmer response

Run an offline Access Method Services job to delete or rename the error BSDS and define a new BSDS with the same name. Re-enter the RECOVER BSDS command to reestablish dual BSDS mode.

CSQJ306I
DUAL BSDS MODE ALREADY ESTABLISHED

Explanation

The **RECOVER BSDS** command was issued, but the queue manager was already in dual BSDS mode.

System action

The command is ignored.

CSQJ307I
LOG INITIALIZED IN SINGLE BSDS MODE

Explanation

The **RECOVER BSDS** command was issued, but the queue manager was initialized in single BSDS mode.

System action

Processing of the command is terminated. The queue manager continues in single BSDS mode.

CSQJ308I
LOG NOT OFFLOADED FOR ARCHIVE LOG COMMAND, ARCHIVING IS OFF

Explanation

The **ARCHIVE LOG** command was issued, but archiving is off (that is, **OFFLOAD** is set to `'NO'` in the CSQ6LOGP system parameters).

System action

The current active log data set is not offloaded. However, it is truncated and logging continues using the next active log data set.

CSQJ309I
QUIESCING FOR ARCHIVE LOG COMMAND WITH WAIT(YES) STARTED FOR MAXIMUM OF *xxx* SECONDS

Explanation

An **ARCHIVE LOG** command with the **MODE(QUIESCE)** and **WAIT(YES)** options has been accepted by the queue manager. The quiesce processing has commenced.

WAIT(YES) means that quiesce processing will be synchronous to the user; that is, the user can enter additional commands, but they will not be processed until the quiesce processing has ended.

System action

The queue manager attempts to stop all updates to IBM MQ resources within the time period specified in the message. Users and jobs using the queue manager are allowed to reach a point of consistency (commit point) before being blocked from further update activity. Users and jobs are suspended until they are released by the queue manager following the initiation of the offload processing. If the queue manager can effectively block all users from performing updates before the maximum specified time, the offload is initiated immediately, and normal processing is resumed.

This message will be followed by message CSQJ311I or CSQJ317I.

CSQJ310I
QUIESCING FOR ARCHIVE LOG COMMAND WITH WAIT(NO) STARTED FOR MAXIMUM OF *xxx* SECONDS

Explanation

An **ARCHIVE LOG** command with the **MODE(QUIESCE)** and **WAIT(NO)** by the queue manager. The quiesce processing has commenced.

WAIT(NO) means that quiesce processing will be asynchronous to the user; that is, control will be returned to the invoker as soon as the quiesce task has been started. Thus, the queue manager will accept, and process, any new commands while the quiesce task is running.

System action

The queue manager attempts to stop all updates to IBM MQ resources within the time period specified in the message. Users and jobs using the queue manager are allowed to reach a point of consistency (commit point) before being blocked from further update activity. Users and jobs are suspended until they are released by the queue manager following the initiation of the offload processing. If the queue manager can effectively block all users from performing updates before the maximum specified time, the offload is initiated immediately, and normal processing is resumed.

This message will be followed by message CSQJ311I or CSQJ317I.

CSQJ311I
csect-name LOG ARCHIVE (OFFLOAD) TASK INITIATED

Explanation

A user-initiated **ARCHIVE LOG** command has been accepted by the queue manager. A task to archive (offload) the active log data set has been started.

System action

The current active log data sets will be truncated and switched to the next available active log data sets. The task has been started will archive the active log data sets asynchronously, allowing the queue manager to continue processing.

This message will be followed by the CSQJ312I message if the **MODE(QUIESCE)** option was used with the **ARCHIVE LOG** command.

CSQJ312I
ARCHIVE LOG QUIESCE ENDED. UPDATE ACTIVITY IS NOW RESUMED

Explanation

An **ARCHIVE LOG** command with the **MODE(QUIESCE)** option was processed by the queue manager. As part of the **MODE(QUIESCE)** processing, an attempt was made to stop all new update activity against IBM MQ resources. This message signals the end of the quiesce processing, and the resumption of normal activity for all users and jobs which were blocked during the quiesce period.

This message will follow the CSQJ311I message or CSQJ317I message.

System action

The queue manager has now resumed all normal activity for all users and jobs which were blocked during the quiesce period.

CSQJ314E
'*kwd1*' requires '*kwd2*' to be specified too

Explanation

A command was entered that specified the *kwd1* keyword. However, use of this keyword requires that the *kwd2* keyword is also used.

System action

Processing for the command is terminated.

CSQJ315I
STOP QMGR MODE(FORCE) IN PROGRESS

Explanation

An attempt was made to issue an **ARCHIVE LOG** command when a **STOP QMGR MODE(FORCE)** command was already in progress.

System action

Command processing will terminate for the **ARCHIVE LOG** command. The **STOP QMGR MODE(FORCE)** processing will continue.

CSQJ316I
SYSTEM QUIESCE ALREADY IN PROGRESS

Explanation

An **ARCHIVE LOG** command with the **MODE(QUIESCE)** option or a **SUSPEND QMGR LOG** command was issued when a system quiesce was already in progress. The system quiesce could be the result of processing by another **ARCHIVE LOG MODE(QUIESCE)** command, or by a **STOP QMGR MODE(QUIESCE)** command.

System action

Command processing will terminate. The system quiesce currently in progress will continue.

CSQJ317I
QUIESCE PERIOD EXPIRED WITH *nn* OUTSTANDING URS AT *time*. ARCHIVE LOG PROCESSING TERMINATED

Explanation

An **ARCHIVE LOG MODE(QUIESCE)** command was processed by the queue manager. However, the queue manager was not able to quiesce all update activity in the user-specified quiesce time interval.

System action

This message is for information only. The queue manager determined that *nn* units of recovery did not reach a point of consistency during the quiesce period, and therefore could not be stopped from continuing their associated update processing.

Consequently, the **ARCHIVE LOG** processing will be terminated. The current active log data sets will not be truncated, and will not be switched to the next available active log data sets. The log archive (offload) task will not be created. All jobs and users suspended during the quiesce will be resumed, and normal update activity against IBM MQ resources will be commenced.

This message will be followed by the CSQJ312I message.

System programmer response

You must decide whether the outstanding (non-quiesced) units of recovery represent significant work.

Each user on the system has a unit of recovery if they are modifying IBM MQ resources. Units of recovery are also created by the queue manager itself for internal processing. Because the purpose of the **MODE(QUIESCE)** option is to have all units of recovery reach a point of consistency (commit point) before the active log data set is truncated and offloaded, determine all outstanding non-queued jobs and users by using **DISPLAY THREAD** and the z/OS command **DISPLAY ACTIVE,LIST**.

Note that units of recovery might be outstanding due to lock contention between a user or job that holds a resource (and has reached a point of consistency), and a user or job that wants a lock (and therefore cannot reach a point of consistency).

Before resubmitting the **ARCHIVE LOG** command with the **MODE(QUIESCE)** option, either:

- Wait until the threads have been deallocated
- Wait until the queue manager is less busy
- Force the offending threads to terminate
- Use the TIME option to override and extend the maximum quiesce time period specified in the system parameters
- If having all units of recovery reach a point of consistency in the active log is no longer critical, issue the **ARCHIVE LOG** command without the **MODE(QUIESCE)** option

Note: If you decide to use the **ARCHIVE LOG** command without the **MODE(QUIESCE)** option, the active log data sets will be truncated without regard to quiescing activity on the queue manager. If the resulting archive log data set is used for recovery, it is possible that some units of recovery might be found to be in-flight, in-backout, in-commit, or in-doubt during queue manager initialization.

If expiration of the quiesce period before all units of recovery reach a consistent point is a problem, you might have to adjust the **QUIESCE** value in the CSQ6ARVP system parameters. For more information, see Using CSQ6ARVP.

CSQJ318I
ARCHIVE LOG COMMAND ALREADY IN PROGRESS

Explanation

An attempt was made to issue an **ARCHIVE LOG** command when another **ARCHIVE LOG** command was already in progress.

System action

Command processing will terminate. The **ARCHIVE LOG** command currently in progress will continue.

CSQJ319I
csect-name CURRENT ACTIVE LOG DATA SET IS THE LAST AVAILABLE ACTIVE LOG DATA SET. ARCHIVE LOG PROCESSING WILL BE TERMINATED

Explanation

The **ARCHIVE LOG** command was rejected because the current active log is the last available active log data set. To process the command when these conditions exist would cause the queue manager to exhaust its available active log resources and immediately halt processing.

System action

Processing for the command is terminated.

If the situation is not corrected, the queue manager will issue the CSQJ110E message (if it has not already done so) when the available active log data space reaches critically low levels. Ultimately, message CSQJ111A will be issued when the available active log data space is exhausted, and processing will stop until active log space is made available.

System programmer response

To clear this condition, steps must be taken to complete other waiting offload tasks. Once another active log is made available (re-usable) by completing the offload process for it, the command processing for the current active log can proceed.

Perform a display request to determine the outstanding requests related to the log offload process. Take the necessary action to satisfy any requests, and permit offload to continue.

If offload does not complete normally, or cannot be initiated, either correct the problem that is causing the offload problem, or consider whether there are sufficient active log data sets. If necessary, additional log data sets can be added dynamically using the **DEFINE LOG** command.

Possible causes for the shortage of active log data space are:

- Excessive logging. For example, there is a lot of persistent message activity.
- Delayed or slow offloading. For example, failure to mount archive volumes, incorrect replies to offload messages, or slow device speeds.
- Excessive use of the **ARCHIVE LOG** command. Each invocation of the command causes the queue manager to switch to a new active log data set. Excessive use could consume the available active log data space if the resulting offloads were not processed in a timely manner.
- Offloads unsuccessful.
- Insufficient active log space.

CSQJ320E
csect-name UNABLE TO PROCESS LOG TRUNCATION REQUEST DUE TO INTERNAL ERROR. (ERROR DATA=*ddd*)

Explanation

While processing an **ARCHIVE LOG** command, an internal request was made of the log buffer output routine to force-write the log buffers and to truncate and switch the active log to the next available active log data sets.

System action

Processing for the command is terminated.

System programmer response

This is an internal error detected by the queue manager. The error might be caused by an unrelated error in the log buffer writer component (CSQJWxxx), by a **STOP QMGR MODE(FORCE)** command, or by abnormal termination. See any messages that precede this message.

CSQJ321E
UNABLE TO CONTINUE ARCHIVE LOG QUIESCE DUE TO INTERNAL ERROR. ARCHIVE LOG PROCESSING TERMINATED

Explanation

An **ARCHIVE LOG** command with the **MODE(QUIESCE)** option was processed by the queue manager. As part of the MODE(QUIESCE) processing, an attempt was made to stop all new update activity against IBM MQ resources. During the processing, an internal error occurred.

System action

The **ARCHIVE LOG MODE(QUIESCE)** processing is terminated. This message will be followed by message CSQJ312I after all users and jobs quiesced by the **MODE(QUIESCE)** processing are resumed.

System programmer response

This error is an internal error detected by the queue manager. Retry the **ARCHIVE LOG MODE(QUIESCE)** command. If the error persists, the active log data sets can be switched using the **ARCHIVE LOG** command without the **MODE(QUIESCE)** option.

CSQJ322I
DISPLAY parm-type report ...

Explanation

This message is part of the response to the **DISPLAY** and **SET** *parm-type* commands (where *parm-type* is **SYSTEM**, **LOG**, or **ARCHIVE**). It provides information about the corresponding system parameters. For example:

```
Parameter  Initial value  SET value
LOGLOAD      500000       400000
CMDUSER      CSQOPR
EXCLMSG      X500,X501,X528,X208,
             X519,X599
End of parm-type report
```

where:

LOGLOAD
was set in CSQ6SYSP and changed using the **SET SYSTEM LOGLOAD** command.

CMDUSER
was set in CSQ6SYSP and has not been changed.

EXCLMSG
was set to the default in CSQ6SYSP, and has been changed using the **SET SYSTEM EXCMSG** command.

System action

Processing continues.

CSQJ325I
ARCHIVE tape unit report ...

Explanation

This message is part of the response to the **DISPLAY** and **SET ARCHIVE** commands. It provides information about tape units used for archive logging, as follows:

```
Addr St CorrelID VolSer DSName  addr st correlid volser dsname|  End of tape unit report
```

where:

addr
The physical address of a tape unit allocated to read the archive log.

st
The status of the tape unit:

B
Busy, actively processing an archive log data set.

P
Premount, active and allocated for premounting.

A
Available, inactive and waiting for work.

*****
Unknown.

correlid
The correlation ID associated with the user of the tape being processed; '*****' if there is no current user.

volser
The volume serial number of the tape that is mounted.

dsname
The data set name on the tape volume that is being processed or was last processed.

If no tape units are allocated, the list is replaced by:

```
No tape archive reading activity
```

System action

Processing continues.

CSQJ330I
ARCHIVE LOG VOLUMES required for connection-ID *xxxx*, correlation-ID *yyyyyy*:

Explanation

This message lists the names of the archive log volumes needed by the indicated correlation ID for the given connection ID. The archive log volumes are listed with a maximum of six on each line. It is generated automatically by the archive read process at the first archive log tape mount for that correlation ID. The connection ID is an identifier representing the connection name used to establish the thread; the correlation ID is an identifier associated with a specified thread, such as a job name.

A volume name prefixed with an '*' signifies that the data on the archive log volume is also mapped by an active log data set. As such, the volume might not be required for the read process, because the data is read from the active log if possible.

The following is an example of the output produced by message CSQJ330I:

```
CSQJ330I: ARCHIVE LOG VOLUMES required for connection-ID xxxx,
 correlation-ID yyyyyy: volume1, volume2, volume3, volume4, volume5, volume6 End of ARCHIVE
LOG VOLUMES report
```

System action

Processing continues.

CSQJ334E
Parameter value is unacceptable for '*kwd*'

Explanation

The parameter value specified is not an acceptable value for the named keyword, or is incompatible with values set for other keywords.

System action

Processing for the command is terminated.

CSQJ335E
Invalid command syntax

Explanation

No keywords or an unacceptable combination of keywords was specified on a command.

System action

Processing for the command is terminated.

CSQJ337I
parm-type parameters set

Explanation

The **SET** command completed successfully, setting system parameter values for the indicated *parm-type* (**SYSTEM**, **LOG**, or **ARCHIVE**).

CSQJ364I
IMS Bridge facility suspended for XCFGNAME=*gname* XCFMNAME=*mname*

Explanation

This is issued as part of the response to a **DISPLAY SYSTEM** command if the IBM MQ-IMS Bridge facility to the partner IMS system identified by *gname* and *mname* is suspended.

System programmer response

Use the **RESUME QMGR FACILITY(IMSBRIDGE)** command when ready to resume the IBM MQ-IMS Bridge.

CSQJ365I
Db2 connection suspended

Explanation

This is issued as part of the response to a **DISPLAY SYSTEM** command if the connection to Db2 is suspended.

System programmer response

Use the **RESUME QMGR FACILITY(DB2)** command when ready to resume the connection to Db2.

CSQJ366I
Logging already suspended

Explanation

A **SUSPEND QMGR LOG** command was issued, but logging was already suspended by a previous command.

System action

The command is ignored.

CSQJ367I
Queue manager stopping

Explanation

A **SUSPEND QMGR LOG** command was issued, but the queue manager is stopping.

System action

The command is ignored.

CSQJ368I
Logging not suspended

Explanation

A **RESUME QMGR LOG** command was issued, but logging was not suspended.

System action

The command is ignored.

CSQJ369E
csect-name Failure while suspending logging

Explanation

A **SUSPEND QMGR LOG** command was issued, but it terminated abnormally.

System action

The command is ignored, and logging is not suspended.

System programmer response

Verify the command entry, and reissue the command. If it fails again, collect the items listed in the Problem Determination section, and contact IBM Support.

CSQJ370I
LOG status report ...

Explanation

This message is part of the response to the **DISPLAY** and **SET LOG** commands. It provides information about the status of the log data sets, as follows:

```
Copy %Full zHyperWrite Encrypted DSName
1    k     p           e         dsname
2    k     p           e         dsname
Copy zHyperLink
1    h
2    h
Restarted at date time using RBA=sss
Latest RBA=rrr
Offload task is xxx
Full logs to offload - m of n
```

where:

1, 2
Information for the current active log copy 1 and copy 2 data sets.

k
The percentage of the active log data set that has been used.

p
Indicates whether this data set is zHyperWrite capable or not.

NO
This log data set is not zHyperWrite capable.

CAPABLE
This log data set is zHyperWrite capable. If the **ZHYWRITE** system parameter has been set to YES, then log writes will be made with zHyperWrite enabled.

YES
All writes to this data set are performed with zHyperWrite enabled.

From IBM MQ 9.4.0:

- If the logs are capable and the value is set to **ZHYWRITE(YES)**, *p* would show the value YES.
- If the logs are not capable and the value is set to **ZHYWRITE(YES)**, *p* would show the value YES.
- If the value is set to **ZHYWRITE(NO)**, the value of *p* reflects the capability of the logs as it did prior to IBM MQ 9.4.0.

e
Indicates whether the data set is encrypted or not.

NO
This data set is not encrypted.

YES
This data set is encrypted.

dsname
The data set name of the active log data set. If the copy is not currently active, this is shown as `Inactive`.

h
Indicates whether this data set is zHyperLink capable or not.

NO
This log data set is not zHyperLink capable.

CAPABLE
This log data set is zHyperLink capable. If the **ZHYLINK** system parameter has been set to YES, then log writes will be attempted with zHyperLink.

YES
All writes to this data set are attempted with zHyperLink.

- If the logs are capable and the value is set to **ZHYLINK(YES)**, *h* would show the value YES.
- If the logs are not capable and the value is set to **ZHYLINK(YES)**, *h* would show the value YES.
- If the value is set to **ZHYLINK(NO)**, the value of *h* reflects the capability of the logs.

date time
The time that the queue manager was started.

sss
The RBA from which logging began when the queue manager was started.

rrr
The RBA of the most recently written log record. If logging is suspended, this line is replaced by

```
Logging suspended at RBA=rrr
```

xxx
The status of the offload task, which can be:

BUSY, allocating archive data set
This could indicate that a tape mount request is pending.

BUSY, copying BSDS
Copying the BSDS data set.

BUSY, copying active log
Copying the active log data set.

BUSY
Other processing.

AVAILABLE
Waiting for work.

m*, *n
The number of full active log data sets that have not yet been archived, and the total number of active log data sets.

System action

Processing continues.

CSQJ372I
Logging suspended for *qmgr-name* at RBA=*rrr*

Explanation

This is issued in response to a **SUSPEND QMGR LOG** command if it completed successfully.

It is also issued in response to other commands if logging is suspended, indicating that the command cannot be processed while logging is suspended.

System action

All log update activity is suspended for the queue manager named. *rrr* is the RBA of the last log record written.

For commands other than **SUSPEND QMGR LOG**, the command is ignored.

System programmer response

Use the **RESUME QMGR LOG** command when ready to resume logging.

CSQJ373I
Logging resumed for *qmgr-name*

Explanation

The **RESUME QMGR LOG** command completed successfully.

System action

All log update activity is resumed for the queue manager named.

CSQJ401E
RECORD NOT FOUND - *rrr*

Explanation

An attempt was made to read the *rrrr* record from the BSDS. In doing so, the read routine (CSQJU01B) could not find the record.

This is not necessarily an error; for example, if you have never used CSQJU003 CRESTART, there will not be any CRCR records, so you will get this message from CSQJU004 for the RESTART CONTROL records.

System action

Utility processing continues.

CSQJ404E
kwd NOT ALLOWED FOR *oper* OPERATION

Explanation

An invalid keyword was used during the *oper* operation.

System action

The current utility processing is terminated.

CSQJ405E
KEYWORDS *kwd1* AND *kwd2* CANNOT BOTH BE SPECIFIED

Explanation

Keywords *kwd1* and *kwd2* cannot appear on the same control statement.

System action

The current utility processing is terminated.

CSQJ406E
EITHER KEYWORD *kwd1* OR *kwd2* MUST BE SPECIFIED

Explanation

A required keyword was not used on the control statement. Use either *kwd1* or *kwd2* with that control statement type.

System action

The current utility processing is terminated.

CSQJ407E
NO VALID CHECKPOINT RBA FOUND

Explanation

After completing its search through the resource manager status table and the checkpoint queue, no valid checkpoint RBA was found within the specified range.

System action

The current utility processing is terminated.

System programmer response

The last 100 checkpoints are recorded in the BSDS, including the log STARTRBA and log ENDRBA of the checkpoint range. The utility attempts to locate a valid checkpoint in the range. In this case the utility was unsuccessful in finding a valid checkpoint.

Use the Print Log Map Utility (CSQJU004) to determine the valid RBA ranges, and rerun the job with a suitable RBA specification.

CSQJ408I
CHECKPOINT RBA FOUND, RBA=*rba*, TIME=*date time*

Explanation

After completing its search through the resource manager status table and the checkpoint queue, *rba* was the most recent checkpoint RBA in the specified range, and *date time* was the time of the checkpoint.

System action

Utility processing continues.

CSQJ409E
I/O ERROR DURING READ PROCESSING OF RECORD - *yyy*

Explanation

An input/output error occurred during a READ of a record. *yyy* specifies the record in question.

System action

The current utility processing is terminated. This message is accompanied by message CSQJ212E.

System programmer response

Determine the cause of the error based on the error status information provided in message CSQJ212E.

CSQJ410E
I/O ERROR DURING WRITE PROCESSING OF RECORD - *yyy*

Explanation

An input/output error occurred during a WRITE of a record. *yyy* specifies the record in question.

System action

The current utility processing is terminated. This message is accompanied by message CSQJ213E.

System programmer response

Determine the cause of the error based upon the error status information provided in message CSQJ213E.

CSQJ411I
CRESTART CREATE FOR CRCRID=*yyyy*, DDNAME=*ddd*

Explanation

A CRESTART CREATE request has just completed. *yyyy* is the restart control record hexadecimal identifier and *ddd* is the BSDS data set (SYSUT1 or SYSUT2) associated with the request.

System action

Current utility processing continues.

System programmer response

Note the record identifier for future reference.

CSQJ412E
RESTART CONTROL RECORD NOT FOUND IN BSDS

Explanation

A CRESTART CANCEL keyword was specified but the conditional restart control record does not exist in the BSDS data set.

System action

Current utility processing is terminated.

System programmer response

None necessary, if CANCEL was the intended action. Otherwise, correct the control statement and rerun the utility.

CSQJ413E
INVALID LOG RANGE SCOPE OR CHECKPOINT SPECIFIED

Explanation

The values specified through the **STARTRBA** and **ENDRBA** keywords are invalid.

System action

Current utility processing is terminated.

System programmer response

Ensure that the log range values are correct and correspond to the other log range values either specified or defaulted. The **STARTRBA** must be less than or equal to the **ENDRBA**.

CSQJ414I
COLD START WILL RESULT FROM THIS RESTART CONTROL RECORD. FORWARD AND BACKOUT SET TO NO

Explanation

STARTRBA and ENDRBA are equal. A cold start will result if this restart control record is used during restart. No forward or backout processing will be performed.

System action

CRESTART processing continues.

System programmer response

No additional actions are needed if a cold start of the queue manager is required. If a cold start is not required, reissue the CRESTART and either CANCEL the current restart control record, or CREATE a new restart control record.

CSQJ415E
ENDRBA=*rba* IS INVALID, MUST BE A MULTIPLE OF 4K

Explanation

The specified ENDRBA at *rba* is not a multiple of 4K.

System action

CRESTART processing is terminated.

System programmer response

Correct the ENDRBA value on the CRESTART statement and rerun the utility.

CSQJ416I
WARNING - BSDS UTILITY TIME STAMP MISMATCH DETECTED. PROCESSING CONTINUES

Explanation

As a result of a change log inventory update, it was discovered that the SYSUT1 BSDS and SYSUT2 BSDS time stamps are unequal. Their inequality indicates the possibility of a BSDS mismatch.

System action

Current utility processing continues.

System programmer response

Run the print log map utility against the SYSUT1 BSDS and SYSUT2 BSDS. Determine if each BSDS is current. If each BSDS is current, this warning can be ignored. If either BSDS is not current, delete the obsolete data set and define a replacement data set, then copy the current BSDS into the replacement data set.

CSQJ417E
REQUIRED *xxxx* PARAMETER FOR *oper* OPERATION IS MISSING

Explanation

Required parameter *xxxx* for a log utility operation was missing from the log utility control statement. The attempted operation is *oper*.

System action

The log utility *oper* operation does not perform its function. All subsequent log utility control statements are processed. A nonzero return code is issued by the utility.

System programmer response

Add the missing parameter to the control statements associated with the specified operation and rerun the utility.

CSQJ418I
NOTREUSABLE ACTIVE LOG DELETED FROM THE BSDS LOG INVENTORY, STARTRBA=*sss* ENDRBA=*ttt*

Explanation

The data set name specified on the DSNAME parameter of the change log inventory utility DELETE statement was a NOTREUSABLE active log.

System action

The change log inventory utility processing continues. It will terminate with a return code of 4.

System programmer response

No additional actions are required if you want to delete a NOTREUSABLE active log. If not, re-create the deleted log by using the NEWLOG statement with the RBA values specified in the warning message.

CSQJ421I
CRESTART CANCEL FOR CRCRID=*yyyy*, DDNAME=*ddd*

Explanation

A CRESTART CANCEL request has just completed. *yyyy* is the restart control record hexadecimal identifier and *ddd* is the BSDS data set (SYSUT1 or SYSUT2) associated with the request.

System action

Current utility processing continues.

System programmer response

Note the record identifier for future reference.

CSQJ425E
INVALID VALUE OR FORMAT FOR *xxxx* PARAMETER (YYYYDDDHHMMSST)

Explanation

The *xxxx* parameter contains an incorrect value or incorrect format for the date and time.

System action

The current utility is terminated.

System programmer response

Correct the control statement and rerun the utility.

CSQJ426E
ENDTIME VALUE CANNOT BE LESS THAN STARTIME VALUE

Explanation

The **STARTIME** and **ENDTIME** parameters specify a time range. Therefore, the **ENDTIME** value must be equal to or greater than **STARTIME** value.

System action

The current utility is terminated.

System programmer response

Correct the control statement and rerun the utility.

CSQJ427I
CHECKPOINT RECORD ADDED TO QUEUE

Explanation

The checkpoint record specified has been added to the checkpoint queue in the BSDS.

System action

Processing continues.

CSQJ428I
CHECKPOINT RECORD DELETED FROM QUEUE, STARTRBA= *ssss* ENDRBA=*ttt*

Explanation

The checkpoint record specified has been deleted from the checkpoint queue in the BSDS. *sss* and *ttt* was the RBA range indicated in the deleted checkpoint record.

System action

Processing continues.

CSQJ429E
RBA RANGE CONFLICTS WITH EXISTING CHECKPOINT RECORD RBA RANGE

Explanation

The specified RBA range for the new checkpoint record either exists, or overlaps an existing RBA range in the checkpoint queue in the BSDS.

System action

The current utility is terminated.

System programmer response

Run the print log map utility against the SYSUT1 BSDS and SYSUT2 BSDS. Determine the correct RBA range, correct the **STARTRBA** and **ENDRBA** parameters, and rerun the utility.

CSQJ430E
SPECIFIED ENTRY CANNOT BE ADDED WITHOUT OVERLAYING EXISTING LOWEST ENTRY

Explanation

The specified RBA range for the new checkpoint record is less than the lowest existing entry. The checkpoint queue in the BSDS is currently full and cannot add the new entry without overlaying the lowest entry.

System action

The current utility is terminated.

System programmer response

Run the print log map utility against the SYSUT1 BSDS and SYSUT2 BSDS. Determine the lowest existing entry, either change the STARTRBA and ENDRBA parameters or delete the lowest existing entry and add a new low checkpoint entry, and rerun the utility.

CSQJ431E
STARTRBA SPECIFIED CANNOT BE FOUND IN CHECKPOINT QUEUE

Explanation

The specified **STARTRBA** could not be located in the checkpoint queue in the BSDS.

System action

The current utility is terminated.

System programmer response

Run the print log map utility against the SYSUT1 BSDS and SYSUT2 BSDS. Determine the correct **STARTRBA** value, correct the **STARTRBA** parameter, and rerun the utility.

CSQJ432E
kwd VALUE MUST END WITH ' *xxx*'

Explanation

The value specified for keyword *kwd* is not valid. It must end with '*xxx*'.

System action

The current utility is terminated.

System programmer response

Correct the control statement and rerun the utility.

CSQJ440I
csect-name IBM MQ for z/OS version

Explanation

This message is issued as part of the header to reports issued by the utility programs.

CSQJ443I
csect-name CHANGE LOG INVENTORY UTILITY - *date time*

Explanation

This message is issued as a header to the report issued by the utility program.

CSQJ444I
csect-name PRINT LOG MAP UTILITY - *date time*

Explanation

This message is issued as a header to the report issued by the utility program.

CSQJ445I
csect-name BSDS CONVERSION UTILITY - *date time*

Explanation

This message is issued as a header to the report issued by the utility program.

CSQJ451E
csect-name BSDS CI SIZE NOT CORRECT, DDNAME=*ddd*

Explanation

A data set provided to the BSDS conversion utility is unusable because the CI size is not correct. The CI size of the BSDS must be 4096. The variable *ddd* contains the DD name of the data set..

System action

The current utility is terminated with no action taken.

System programmer response

Ensure that the DD statement refers to a valid BSDS. If the DD name refers to an output data set, delete and redefine the output BSDS, then rerun the utility.

CSQJ452E
csect-name BSDS UTILITY TIMESTAMP MISMATCH DETECTED

Explanation

A mismatch was detected in the timestamps for the SYSUT1 and SYSUT2 BSDS copies during execution of the BSDS conversion utility. This mismatch indicates the possibility that the dual BSDSs are out of sync.

System action

The current utility is terminated with no action taken.

System programmer response

Run the print log map utility (CSQJU004) against each BSDS. From the output, determine which data set is obsolete, delete it, and define a replacement for it. Then copy the remaining data set into the replacement and try the utility again.

If output from the print log map utility for both data sets is similar, delete the data set with the oldest timestamp, then copy the data set with the most recent timestamp into the replacement.

CSQJ453E
csect-name INPUT BSDS NOT IN CORRECT FORMAT, DDNAME=*ddd*

Explanation

The BSDS conversion utility detected that the input BSDS was not in the correct format to be converted. The input BSDS must be in version 1 format. The variable *ddd* contains the DD name of the data set.

System action

The current utility is terminated with no action taken.

System programmer response

Run the print log map utility (CSQJU004) against the BSDS to determine its version. Ensure that the DD statement refers to an input BSDS in version 1 format, then rerun the utility if necessary.

CSQJ454E
csect-name UNRECOGNIZED BSDS RECORD, KEY=*key-value*

Explanation

During conversion of the BSDS, a record was found that is not a known format. The *key-value* is the VSAM KSDS key of the BSDS record that was not recognized.

System action

The current utility is terminated.

System programmer response

To determine the operation that inserted the record into the BSDS, use IDCAMS PRINT and specify this key value. If the record is not needed, delete it then rerun the BSDS conversion.

CSQJ455E
INVALID BSDS CONVERSION

Explanation

This message is issued when a utility, attempting to access the BSDS data set, encounters an invalid BSDS. An invalid BSDS is the result of a failure during a prior attempt to run the BSDS conversion utility.

System action

The current utility is terminated with no action taken.

System programmer response

The procedure for running the BSDS conversion utility involves renaming the original BSDS. Restore the BSDS to the original pre-conversion copy, by renaming the data sets, then try the conversion again.

CSQJ456E
xxxx PARAMETER ARGUMENT EXCEEDS MAXIMUM VALUE FOR BSDS VERSION *n*

Explanation

The *xxxx* parameter specifies the name of the parameter with a value that exceeds the maximum that can be specified for a BSDS in version *n* format.

System action

The current utility is terminated.

System programmer response

Correct the parameter argument on the control statement, then rerun the utility.

CSQJ491I
csect-name Log Data Set Preformatter Utility - *date time*

Explanation

This message is issued as a header to the report issued by the utility program.

CSQJ492I
Log data set name = *dsname*

Explanation

This identifies the name of the log data set to be pre-formatted.

CSQJ493I
Log data set is not VSAM

Explanation

The input log data set is not a VSAM data set.

System action

Utility processing is terminated.

System programmer response

Check that the SYSUT1 DD statement and the data set name is specified correctly. Use Access Method Services to define the data set as a VSAM linear data set.

CSQJ494E
VSAM OPEN failed, ACBERRFLG=*ee*

Explanation

Opening the log data set failed with the indicated ACB error code.

System action

Utility processing is terminated if the error code is 128 or more; otherwise processing continues.

System programmer response

See the *z/OS DFSMS Macro Instructions for Data Sets* manual for information about the VSAM error code.

CSQJ495E
VSAM PUT failed, RPLERREG=*ee* reason code=*reason*

Explanation

Writing the log data set failed with the indicated RPL error code and reason code.

System action

Utility processing is terminated.

System programmer response

See the *z/OS DFSMS Macro Instructions for Data Sets* manual for information about the VSAM error code.

CSQJ496I
Log preformat completed successfully, *n* records formatted

Explanation

The active log data set has been pre-formatted successfully.

System action

Utility processing is complete.

CSQJ497I
Log preformat terminated

Explanation

Preformatting the active log data set did not complete successfully.

System action

Utility processing is terminated.

System programmer response

See the preceding error messages for more information.

CSQJ498I
Log data set is not empty

Explanation

The input log data set is not an empty data set.

System action

Utility processing is terminated.

System programmer response

Check that the SYSUT1 DD statement and the data set name is specified correctly. Use Access Method Services to define the data set as a VSAM linear data set.

CSQJ499I

Log data set is larger than 4GB

Explanation

The log preformat utility, CSQJUFMT, detected that the VSAM data set to be formatted is greater than 4 GB in size.

System action

Processing continues. The entire data set will be pre-formatted, but IBM MQ for z/OS log data sets are restricted to a maximum of 4 GB. Any additional space in the data set is not used to hold log data.

If an SMDS is being pre-formatted that is intended to be greater than 4 GB, if it is defined using an SMS data class which has the VSAM extended addressability attribute it will not be restricted to a maximum of 4 GB.

System programmer response

Check that the data set name is specified correctly. Use Access Method Services to define the data set with a maximum size of 4 GB.

CSQJ600E
ZHYLINK (YES) specified but no active logs are zHyperLink capable.

Explanation

The system parameter **ZHYLINK** has been set to YES, but the queue manager detects that none of the active log copies are on zHyperLink capable volumes. The status of zHyperLink log capability is asserted at queue manager start up, and therefore might be stale if the capability changes.

The queue manager issues log writes with zHyperLink enabled. regardless of whether the active log copies are on zHyperLink capable volumes.

See Using zHyperLink with IBM MQ for more information.

System action
Processing continues without zHyperLink being enabled for log writes.

System programmer response
Review the configuration for the active log volumes and the zHyperLink feature. Consider checking zHyperWrite capability.

See Troubleshooting zHyperLink for more information.

CSQJ601E
zHyperLink configuration is inconsistent for active log data sets.

Explanation

The queue manager has detected an inconsistency in the zHyperLink configuration for the active log data sets.

The data sets for each copy of the active log should be consistently configured so that they are capable of being used with zHyperLink.

This means that, either all of the data sets that comprise an active log copy should be on zHyperLink capable volumes, or none of the data sets should be on zHyperLink capable volumes.

System action
Processing continues, but an inconsistent logging rate might be observed due to the inconsistent configuration of the active log volumes.

System programmer response
Review the configuration for the active log volumes.

See Troubleshooting zHyperLink for more information.

CSQJ602I
ZHYLINK(YES) requires ZHYWRITE(YES) to be set.

Explanation
A **SET LOG** command specified the **ZHYWRITE(NO)** parameter, but **ZHYLINK(YES)** is specified on the queue manager.

System action
Processing for the command is ended.

System programmer response
Consider whether you want to switch off zHyperWrite. If you do, you also need to switch off zHyperLink at the same time.

CSQJ603I
ZHYWRITE(YES) set due to ZHYLINK(YES) being set.

Explanation
A **SET LOG** command specified **ZHYLINK(YES)** but **ZHYWRITE(NO)** is specified on the queue manager. **ZHYLINK** requires **ZHYWRITE** to be switched on.

System action
The **ZHYWRITE** parameter is set to YES.

CSQJ604E
Failed to disconnect zHyperLink session for *dsname* RC=*ret*

Explanation
An unexpected error was encountered when disconnecting a zHyperLink session for log dataset *dsname*. The zHyperLink session might be left open.

System action
Processing continues.

System programmer response
Check zHyperLink for any left-over sessions with the command `DS QD,MACH=nnnn-xxxxx,ZHL`, where nnnn is the `scu` and `xxxxx` is the serial number. See Troubleshooting zHyperLink for more information.

If the problem persists, contact IBM Support.

CSQJ605E
There were no zHyperLink writes for data set *dsname*

Explanation
During the life cycle of the active log, no successful zHyperLink writes were recorded whilst **ZHYLINK(YES)** was specified. This could be an indication of a configuration or performance problem in your zHyperLink setup.

System action
Processing continues.

System programmer response
Evaluate the reason or reasons for zHyperLink writes falling back to asynchronous writes.

See Troubleshooting zHyperLink for more information. If you are unable to solve the problem, contact IBM Support.

z/OS *Message manager messages (CSQM...)*

CSQM001E

csect-name MSTR user ID cannot invoke USS callable services

Severity
8

Explanation

The IBM MQ queue manager MSTR address space is running under a user ID that has not been configured with authority to execute callable z/OS UNIX System Services (z/OS UNIX).

In RACF, the user ID requires an OMVS segment with a UID assigned.

System action

This message is issued and the process of z/OS UNIX calls, for reverse DNS host name lookup, are disabled in the MSTR address space.

System programmer response

Refer to Planning your z/OS UNIX environment, where queue manager MSTR and CHIN address spaces require user IDs with OMVS segments defined with a valid UID.

Correct the configuration of the queue manager MSTR address space user ID and restart the queue manager.

CSQM050I

csect-name Intra-group queuing agent starting, TCB=*tcb-name*

Severity
0

Explanation

The intra-group queuing (IGQ) agent was started during the initialization of a queue manager that is in a queue sharing group. The agent uses TCB *tcb-name*.

The IGQ agent handles SYSTEM.QSG.TRANSMIT.QUEUE.

System action

Processing continues. The IGQ agent starts asynchronously.

CSQM051I

csect-name Intra-group queuing agent stopping

Severity
0

Explanation

The intra-group queuing (IGQ) agent is stopping because:

- the queue manager is stopping
- it has retried a failing request repeatedly without success
- it was unable to recover from an abnormal ending

System action

The IGQ agent stops.

System programmer response

If the queue manager is not stopping, investigate the cause of the error as reported in the preceding messages. To restart the IGQ agent, issue an ALTER QMGR command specifying IGQ(ENABLED).

CSQM052I

csect-name Shared channel recovery completed for *qmgr-name*, *n* channels found, *p* FIXSHARED, *r* recovered

Severity
0

Explanation

The queue manager successfully recovered some shared channels that were owned by queue manager *qmgr-name* in the queue sharing group when it or its channel initiator terminated abnormally. This recovery process might occur when:

- another queue manager or its channel initiator terminates abnormally
- the channel initiator is started, for channels that were owned by other queue managers
- the channel initiator is started, for channels that were owned by itself

n channels were found that needed recovery, of which *p* were originally started as FIXSHARED. The number recovered, *r*, might be less than *n* (or even 0) because other active queue managers are also

recovering the channels and because FIXSHARED channels cannot be recovered by another queue manager.

For more information about shared channel recovery, see Shared channels.

System action

Processing continues.

CSQM053E

csect-name Shared channel recovery terminated, DB2 not available

Severity
8

Explanation

Because Db2 is not available or no longer available, the queue manager was unable to recover some shared channels that were owned by a queue manager in the queue sharing group when it or its channel initiator terminated abnormally. This recovery process might occur when:

- another queue manager or its channel initiator terminates abnormally
- the channel initiator is started, for channels that were owned by other queue managers
- the channel initiator is started, for channels that were owned by itself

System action

The recovery process is terminated; some channels might have been recovered, while others have not.

System programmer response

Use the preceding messages on the z/OS console to investigate why Db2 is not available, and resume the connection or restart Db2 if necessary. Any channels that were not recovered will be recovered when the recovery process next runs; alternatively, they can be restarted manually.

CSQM054E

csect-name Shared channel recovery terminated, error accessing DB2

Severity
8

Explanation

Because there was an error in accessing Db2, the queue manager was unable to recover some shared channels that were owned by a queue manager in the queue sharing group when it or its channel initiator terminated abnormally. This recovery process might occur when:

- another queue manager or its channel initiator terminates abnormally
- the channel initiator is started, for channels that were owned by other queue managers
- the channel initiator is started, for channels that were owned by itself

System action

The recovery process is terminated; some channels might have been recovered, while others have not.

System programmer response

Resolve the error reported in the preceding messages. Any channels that were not recovered will be recovered when the recovery process next runs; alternatively, they can be restarted manually.

CSQM055E

csect-name Shared channel recovery terminated, error putting command, MQRC=*mqrc (mqrc-text)*

Severity
8

Explanation

Because there was an error putting a message on the system-command input queue, the queue manager was unable to recover some shared channels that were owned by a queue manager in the queue sharing group when it or its channel initiator terminated abnormally. This recovery process might occur when:

- another queue manager or its channel initiator terminates abnormally
- the channel initiator is started, for channels that were owned by other queue managers
- the channel initiator is started, for channels that were owned by itself

System action

The recovery process is terminated; some channels might have been recovered, while others have not.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqrc* (*mqrc-text* provides the MQRC in textual form), and resolve the error. Any channels that were not recovered will be recovered when the recovery process next runs; alternatively, they can be restarted manually.

CSQM056E

csect-name mqapi-call failed for queue *q-name*, MQRC=*mqrc (mqrc-text)*

Severity
8

Explanation

The indicated IBM MQ API call for the named queue, failed for the specified reason, which might be an IBM MQ reason code (MQRC_) or a signal completion code (MQEC_).

System action

If the queue is SYSTEM.ADMIN.CONFIG.EVENT or SYSTEM.ADMIN.COMMAND.EVENT, processing continues but events are not generated; message CSQM071E follows to show how many event messages have not been generated since the problem first occurred. These messages are generated on the first occurrence of the problem, and at intervals thereafter while the problem persists.

Depending on the queue involved and the type of error, it might continue processing, try the request again at regular intervals until the error is corrected, or terminate.

System programmer response

For more information, see "API completion and reason codes" on page 1151. For information about signal completion codes, see Signaling. Correct the problem with the queue, or use the ALTER QMGR command to disable the events.

CSQM057E

csect-name MQPUT of trigger message failed for queue *q-name*, MQRC=*mqrc (mqrc-text)*

Severity
8

Explanation

The queue manager could not deliver a trigger message to the indicated initiation queue for the specified IBM MQ reason code (MQRC_).

System action

The queue manager attempts to put the trigger message on to the dead-letter queue if one has been defined.

System programmer response

For more information about IBM MQ reason codes, and what action to take to correct the problem with the initiation queue, see "API completion and reason codes" on page 1151.

CSQM058E

csect-name Unable to start channel *channel-name*

Severity
8

Explanation

An attempt was made to start cluster channel *channel-name* because a message was placed on the SYSTEM.CLUSTER.TRANSMIT.QUEUE. If the channel could not be started because of an internal queuing error this message is preceded by CSQM056E. This message is also issued if the queue manager encounters a storage shortage.

System action

The message remains queued on the SYSTEM.CLUSTER.TRANSMIT.QUEUE queue and the original MQPUT completes successfully. If the cluster channel is not already running it is not automatically started.

System programmer response

If required, manually start the channel using the START CHANNEL command. Stopping and restarting the channel initiator or the queue manager, or placing another message on the transmission queue for this cluster destination triggers another START request.

If message CSQM056E is issued because of an internal queuing error, action might be needed to ensure that future start channel requests can be processed correctly.

If there is a lack of storage and the problem persists, you might need to increase the region size used by your queue manager, or you might need to reduce the number of jobs running in your system.

CSQM059E

csect-name Queue *q-name* has incorrect attributes

Severity
8

Explanation

The named queue, used by the intra-group queuing (IGQ) agent, has incorrect attributes. For example, SYSTEM.QSG.TRANSMIT.QUEUE must have attributes USAGE(XMITQ), INDXTYPE(CORRELID), QSGDISP(SHARED).

System action

The IGQ agent retries at regular intervals until the error is corrected.

System programmer response

Redefine the queue with the correct attributes.

CSQM060E

csect-name Cluster cache is full

Severity
8

Explanation

No more space is available in the cluster cache area.

System action

The application call that resulted in the need for more space will fail with MQRC_CLUSTER_RESOURCE_ERROR. Processing continues, and existing users of clustering will be unaffected unless their actions are such as to need more cluster cache space.

System programmer response

The problem may be temporary. If it persists, the queue manager must be restarted; this will cause more space to be allocated for the cluster cache area.

Consider changing the cluster cache type system parameter CLCACHE to dynamic, so that more space for the cache will be obtained automatically as required. (If you are using a cluster workload exit, ensure that it supports a dynamic cluster cache.) For information about the system parameters for the CSQ6SYSP macro, see Using CSQ6SYSP.

CSQM061E

csect-name Cluster workload exit *exit-name* does not support dynamic cache

Severity
8

Explanation

In response to the initialization call (using ExitReason MQXR_INIT), the cluster workload exit returned the value MQCLCT_STATIC in the ExitResponse2 field, indicating that it does not support a dynamic cluster cache.

System action

The cluster workload exit is suppressed.

System programmer response

Either change the cluster cache type system parameter CLCACHE to static, or rewrite the exit to be compatible with a dynamic cache. For information about the system parameters for the CSQ6SYSP macro, see Using CSQ6SYSP.

CSQM062I

csect-name INDXTYPE(*index-type*) not allowed for shared transmission queue *shared-xmitq*

Severity
4

Explanation

A shared transmission queue is a queue that is defined with both USAGE(XMITQ) and QSGDISP(SHARED). To support recovery of messages that are in-doubt after a channel failure, the index type (INDXTYPE) for shared transmission queues must be either NONE or MSGID.

System action

Processing continues.

System programmer response

Modify the INDXTYPE attribute for the shared transmission queue to NONE or MSGID.

CSQM063E

csect-name Specified dead-letter queue name is unacceptable

Severity
4

Explanation

The intra-group queuing (IGQ) agent has attempted to put a persistent message on the dead-letter queue that is defined to the queue manager. The dead-letter queue specified is either SYSTEM.QSG.TRANSMIT.QUEUE or there is no dead-letter queue name specified.

System action

The put of the message to the dead-letter queue does not take place, the get of the message from the SYSTEM.QSG.TRANSMIT.QUEUE is backed out and the intra-group queuing (IGQ) agent goes into retry.

System programmer response

Ensure the queue manager has a dead-letter queue defined which is neither blank nor SYSTEM.QSG.TRANSMIT.QUEUE. Examine the message to determine the reason for its placement on the dead-letter queue.

CSQM064I

csect-name Intra-group queuing agent put messages to dead-letter queue

Severity
4

Explanation

The intra-group queuing (IGQ) agent was unable to deliver some messages to the required destination queue, so has put them on the dead-letter queue.

System action

Processing continues.

System programmer response

Examine the contents of the dead-letter queue. Each message is contained in a structure that describes why the message was put to the queue, and to where it was originally addressed.

CSQM065E

csect-name mqapi-call failed, MQRC=*mqrc (mqrc-text)*

Severity
8

Explanation

The indicated MQ API call failed for the specified reason, which is an IBM MQ reason code *mqrc* (mqrc-text provides the MQRC in textual form).

System action

It is the intra-group queuing (IGQ) agent that issued the call; it was unable to commit or backout a batch of messages for the specified reason. Depending on the type of error, it may retry the request at regular intervals until the error is corrected, or terminate.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about MQ reason codes. Correct the problem if required.

CSQM067E

csect-name Intra-group queuing agent ended abnormally. Restarting

Severity
8

Explanation

The intra-group queuing (IGQ) agent has ended abnormally because a severe error occurred, as reported in the preceding messages.

System action

The IGQ agent attempts to restart a number of times. If it fails persistently, it terminates.

System programmer response

Investigate the reason for the abnormal termination, as reported in the preceding messages.

CSQM068I

csect-name Failed to rebuild *n* retained publications

Severity
4

Explanation

While rebuilding the retained publications, *n* messages were found on the SYSTEM.RETAINED.PUB.QUEUE without any message properties.

System action

The associated retained publications were not rebuilt.

System programmer response

If messages were recently moved to the SYSTEM.RETAINED.PUB.QUEUE, then ensure that the PROPCTL value of the source queue does not result in any message properties being lost.

If no messages were recently moved to the SYSTEM.RETAINED.PUB.QUEUE, then note this message and contact IBM Support.

CSQM070E

csect-name Queue *q-name* available again, *n* events not generated

Severity
4

Explanation

An earlier problem with putting messages on the configuration or command event queue has been corrected. *n* is the number of event messages that have not been generated since the problem first occurred.

System action

Processing continues and event messages for that queue will be generated again.

System programmer response

If the queue is SYSTEM.ADMIN.CONFIG.EVENT, and complete configuration information is required, use the REFRESH QMGR TYPE(CONFIGEV) command to generate events to replace those that were not generated; specify the INCLINT parameter to cover the period when the problem was occurring.

If the queue is SYSTEM.ADMIN.COMMAND.EVENT, a limited number of the missed event messages may be recovered automatically, as reported by message CSQM072I.

CSQM071E

csect-name Queue *q-name* unavailable, *n* events not generated

Severity
8

Explanation

There was an error putting a message on the configuration or command event queue, as reported in the preceding CSQM056E message; *n* is the number of event messages that have not been generated since the problem first occurred.

System action

Processing continues but event messages for that queue are not generated. This message is issued on the first occurrence of the problem, and at intervals thereafter while the problem persists.

System programmer response

Correct the problem with the event queue, or use the ALTER QMGR command to set the CONFIGEV or CMDEV attribute to DISABLED if events are not required.

CSQM072I

csect-name Queue *q-name*, *n* events recovered

Severity
0

Explanation

An earlier problem with putting messages on the command event queue has been corrected. *n* event messages that were not generated have been automatically recovered and generated.

Only a limited number of the missed event messages can be recovered in this way. If *n* is less than the value reported in message CSQM070E, the remaining event messages are lost, and there is no way to recover them.

System action

Processing continues.

CSQM073I

csect-name Loading of durable subscribers started

Severity
0

Explanation

Information about the durable subscribers on a queue manager is stored on the SYSTEM.DURABLE.SUBSCRIBER.QUEUE queue. During the restart of the queue manager the durable subscriptions are remade on the queue manager.

System action

Processing continues.

CSQM074I

csect-name Loading of durable subscribers finished

Severity
0

Explanation

The queue manager has finished reloading all of the durable subscribers.

System action

Processing continues.

CSQM075I

csect-name Consolidation of durable subscribers started

Severity
0

Explanation

Information about the durable subscribers on a queue manager is stored on the SYSTEM.DURABLE.SUBSCRIBER.QUEUE queue. To aid in restart processing and to speed up the time it takes to reload all of the durable subscribers, these messages are consolidated into fewer messages.

System action

Processing continues.

CSQM076I

csect-name Consolidation of durable subscribers finished

Severity
0

Explanation

The queue manager has finished consolidating the messages on the SYSTEM.DURABLE.SUBSCRIBER.QUEUE queue. The processing might be restarted at a later stage if there is a change in the number of durable subscribers.

System action

Processing continues

CSQM077I

csect-name PUBLISH/SUBSCRIBE ENGINE HAS SHUTDOWN

Severity
0

Explanation

The publish/subscribe engine has been shutdown.

System action

The publish/subscribe engine has shutdown.

System programmer response

No action is required if the queue manager is stopping. If the publish/subscribe engine has shutdown because you have disabled it, updating the PSMODE queue manager attribute from the value DISABLED will restart it.

CSQM078E

csect-name Unable to create thread structures for connection-type *connection* from *jobname*, insufficient ACE storage

Severity
8

Explanation

jobname attempted to create a new connection to IBM MQ as the result of issuing the first IBM MQ API call on a new thread. The connection-type is likely to be RRSBATCH.

There was insufficient common storage available to build the control blocks to represent the connection and the connect attempt failed.

There might be a system wide ECSA shortage, or the storage available for creating new queue manager connections might be limited by the ACELIM system parameter.

This message can be seen for CICS and the channel initiator, as well as for RRS applications; for example, Db2 stored procedures and WebSphere Application Server.

System action

IBM MQ API request fails with return code MQRC_STORAGE_NOT_AVAILABLE 2071

Queue manager processing continues

CSQM079I

csect-name Policy access attempt rejected due to incompatible AMS version, jobname *jobname*

Severity
4

Explanation

An incompatible version of Advanced Message Security (AMS), identified by *jobname*, attempted to open the policy queue, SYSTEM.PROTECTION.POLICY.QUEUE.

System action

The request to open the policy queue is rejected.

System programmer response

Update the incompatible version of AMS so it does not attempt to connect to the queue manager. From IBM MQ 8.0, AMS is provided as an integrated feature of IBM MQ for z/OS. For information about how to configure AMS as an integrated feature, see Installing IBM MQ Advanced for z/OS or Installing IBM MQ Advanced for z/OS Value Unit Edition.

CSQM084I

csect-name COMMAND INHIBITED DURING RESTART/TERMINATION

Severity
8

Explanation

A command that will affect a recoverable object was requested either too early in queue manager startup, or too late in termination.

The usual reason for receiving this message is that some prohibited command was issued in the initialization input data set CSQINP1.

System action

Message CSQM085I is also issued and the command is ignored.

System programmer response

Wait until the queue manager is in a state where it is possible to reissue the prohibited commands. If appropriate, remove the command from CSQINP1, and place it in CSQINP2, to ensure that this problem does not recur.

CSQM085I

csect-name ABNORMAL COMPLETION

Severity
8

Explanation

This message is issued with message CSQM084I, and indicates that the command requested has not been actioned.

System action

The command is not actioned.

System programmer response

Wait until the queue manager is in a state where it is possible to use the prohibited commands.

CSQM086E

QUEUE MANAGER CREATE ERROR, CODE=*reason-code*, RESTART UNSUCCESSFUL

Severity
8

Explanation

During restart, the creation of the queue manager object has failed. The reason code is of the form '00D44xxx'.

System action

The queue manager fails to restart.

System programmer response

See “Message manager codes (X'D4')” on page 1015 for an explanation of the reason code, and what action to take. Reissue the START QMGR command to restart the queue manager. If the error persists note this reason code, and contact your IBM support center.

CSQM090E

csect-name FAILURE REASON CODE *reason-code*

Severity
8

Explanation

A command has failed. The reason code is of the form '00D44xxx'. This message is accompanied by one or more other more specific messages, which indicate the reason for the failure.

System action

The command is ignored.

System programmer response

See the explanations of the accompanying messages for more information. See “Message manager codes (X'D4')” on page 1015 for an explanation of the reason code, and what action to take. If the reason code is not one of those listed, make a note of it and contact your IBM support center.

CSQM091E

csect-name FAILURE MQRC=*mqrc (mqrc-text)*

Severity
8

Explanation

A command has failed. The reason code is an IBM MQ reason code. This message is accompanied by one or more other more specific messages, which indicate the reason for the failure.

System action

The command is ignored.

System programmer response

See the explanations of the accompanying messages for more information. Refer to “API completion and reason codes” on page 1151 for an explanation of *mqrc*, (*mqrc-text* provides the MQRC in textual form), and what action to take.

CSQM092I

csect-name keyword(*value*) VALUE INVALID OR OUT OF RANGE

Severity
8

Explanation

Either:

- A keyword was entered that takes a bounded numeric value but the value specified is outside the bounds.

- A keyword was entered that takes a pair of numeric values defining a range, but only one value is specified or the values are not in ascending order.

System action

The command is ignored.

System programmer response

Reissue the command with the parameter specified correctly. For more information about the command, see MQSC commands.

CSQM093I

csect-name keyword(value) NAME CONTAINS INVALID CHARACTERS

Severity
8

Explanation

A name was specified that contains one or more invalid characters. See MQSC commands for information about validation required for the name in question to correct this.

System action

The command is ignored.

System programmer response

Reissue the command with the correct name. For more information about the command, see MQSC commands.

CSQM094I

csect-name keyword(value) WAS NOT FOUND

Severity
8

Explanation

A command was issued that refers to an object that does not exist. That is, no object could be found with the specified name and type (and subtype, for queues and channels) and with any disposition in the queue sharing group.

System action

The command is ignored.

System programmer response

Check that you specified the correct name for the object, and the correct subtype (for queues and channels). If a queue sharing group is in use, check that Db2 is available and not suspended. Define the object if necessary.

Note:

1. If you are dealing with a queue or channel object, an object of the same name, but of a different subtype, might already exist.
2. Remember that the object might have recently been deleted by someone else, or from another queue manager in the queue sharing group.

CSQM095I

csect-name keyword(value) existing-disposition ALREADY EXISTS

Severity
8

Explanation

A DEFINE command was issued, but an object of that type with the specified name already exists, although it might not necessarily have the same subtype, or the same disposition in the queue sharing group. (You cannot have a locally-defined object and a local copy of a group object with the same name; for local queues, you cannot have a shared queue with the same name as a queue with any other disposition.) Where applicable, *existing-disposition* identifies the queue sharing group disposition of the existing object.

System action

The command is ignored.

System programmer response

Reissue the command with another name or with the REPLACE option, or use the existing object, as appropriate.

CSQM096I

csect-name keyword(value) NAME HAS INVALID LENGTH

Severity
8

Explanation

A name was specified that is of an incorrect length.

System action

The command is ignored.

System programmer response

Reissue the command with a name of the correct length. For more information about the command, see MQSC commands.

CSQM097I

csect-name keyword(value) NAME CANNOT BE COMPLETELY BLANK

Severity
8

Explanation

A name was specified that is blank. This is not allowed.

System action

The command is ignored.

System programmer response

Reissue the command with a non-blank name. For more information about the command, see MQSC commands.

CSQM098I

csect-name keyword(value) FIELD TOO LONG

Severity
8

Explanation

Either a numeric or character parameter was specified but it is too long, or (if *value* is blank) a list of character parameters was specified with a total length that is too long.

System action

The command is ignored.

System programmer response

Reissue the command with the correct field length. For more information about the command, see MQSC commands.

CSQM099I

csect-name keyword(value) NAME IN USE AS A DIFFERENT TYPE

Severity
8

Explanation

An object was specified as one particular subtype, but it already exists as another subtype, although it might not necessarily have the same disposition in the queue sharing group. (You cannot have a locally-defined object and a local copy of a group object with the same name; for local queues, you cannot have a shared queue with the same name as a queue with any other disposition.)

System action

The command is ignored.

System programmer response

Reissue the command with the correct name and subtype. For more information about the command, see MQSC commands.

CSQM100I

csect-name keyword(value) VALUE INVALID OR OUT OF RANGE

Severity
8

Explanation

A value is invalid or out of range. This could be because:

- A keyword was entered that takes a series of character values, but the value specified is not one of them.
- A keyword was entered that takes a series of character values, but the value specified is not valid for the particular subtype of object.
- A keyword was entered that takes a bounded numeric value, but the value specified is outside the bounds.
- A keyword was entered that takes a character or hexadecimal value, but the value specified is invalid for that keyword.

System action

The command is ignored.

System programmer response

Reissue the command with the parameter specified correctly. For more information about the command, see MQSC commands.

CSQM101I

csect-name keyword(value) IS CURRENTLY IN USE

Severity
8

Explanation

The object specified is in use. This could be because:

- It is open through the API.
- A trigger message is presently being written to it.

- It is in the process of being deleted.
- When it is a storage class, there is a queue defined as using the storage class, and there are messages currently on the queue.
- When it is a CF structure, there is a queue defined as using the CF structure, and there are messages currently on the queue or the queue is open.
- When altering the index type of a queue, the necessary conditions regarding messages and uncommitted activity are not satisfied.
- When altering the default transmission queue, the old queue is currently being used as a transmission queue by default.
- Although the FORCE option was specified to overcome the object being open through the API, the object was created with a previous version of IBM MQ.
- There is no connection from the queue manager to the structure.

System action

The command is ignored.

System programmer response

Either:

- Wait until the object has been closed or deleted.

  Note: MCAs for receiver channels, or the intra-group queuing (IGQ) agent, can keep the destination queues open for a while even when messages are not being transmitted, and so such queues might appear to be in use.
- Wait until all the queues that use a storage class are empty
- Wait until the queue is empty
- Wait until use of the queue as a default transmission queue has ended

It is not possible to use the FORCE option of the ALTER command to overcome the situations that cause this message.

For more information about the command, see MQSC commands.

CSQM102E

csect-name SSLCIPH *sslciph* IS A WEAK OR BROKEN CIPHERSPEC

Severity
8

Explanation

A channel could not be defined or altered, because the specified SSLCIPH parameter contains a CipherSpec that is potentially insecure.

System action

The named channel is not defined or altered

System programmer response

Examine the CipherSpec specified in the SSLCIPH parameter and consider using a more secure CipherSpec.

If you want to re-enable the use of weak or deprecated CipherSpecs, see Enabling deprecated CipherSpecs on z/OS.

There are alternative mechanisms that can be used to forcibly re-enable weak CipherSpecs, and SSLv3 support, if the Data Definition change is unsuitable. Contact IBM Service for further information.

Attention: Re-enabling CipherSpecs in this manner leaves systems exposed to possible security problems. You should use CipherSpecs that use only the TLS protocol, rather than SSLv3.

CSQM103I

csect-name keyword(value) QSGDISP(*disposition*) HAS MESSAGES ASSOCIATED WITH IT

Severity
8

Explanation

A local queue specified for deletion has messages associated with it, and the DELETE request did not include the PURGE option.

System action

The command is ignored.

System programmer response

Either delete the local queue when it is empty, or reissue the request specifying the PURGE option. If the queue is a local copy of a group object, you must issue the request specifying PURGE explicitly for the local copy; specifying PURGE on the request to delete the group object has no effect.

CSQM104I

csect-name keyword(value) FLAGGED FOR DEFERRED DELETION

Severity
8

Explanation

A local dynamic queue specified on a DEFINE, ALTER, or DELETE request has been flagged for deferred deletion because it was found to be in use at the time of deletion.

System action

The queue is no longer available to new users, and will be deleted when all existing users of it have relinquished access.

CSQM105I

csect-name '*keyword*' VALUE IS SAME AS QALIAS NAME

Severity
8

Explanation

An attempt was made to DEFINE or ALTER an alias queue so that the queue itself was named on the TARGQ keyword. Unless the queue is a cluster queue, this is not allowed because an alias queue can only resolve to a local or remote queue.

System action

The command is ignored.

System programmer response

Reissue the command with a different name for the TARGQ keyword.

CSQM106I

csect-name DEFXMITQ(*q-name*) IS NOT ALLOWED

Severity
8

Explanation

The specified queue is not allowed to be used as the default transmission queue because it is reserved for use exclusively by clustering.

System action

The command is ignored.

System programmer response

Reissue the command with a different DEFXMITQ name.

CSQM107I

csect-name STGCLASS ACTIVE OR QUEUE IN USE

Severity
8

Explanation

A request to ALTER or DEFINE REPLACE a local queue involving a change to the STGCLASS field is not allowed because there are messages on the queue, or other threads have the queue open.

System action

The command is ignored.

System programmer response

If there are messages on the queue, you must remove them before changing the storage class.

Note: If you remove all the messages from the queue, there might be a short delay before the command can be processed successfully.

If other threads have the queue open, wait until they have closed the queue before reissuing the command.

CSQM108I

csect-name keyword(value) NOT ALLOWED, INCOMPATIBLE NAME AND TYPE

Severity
8

Explanation

An attempt was made to issue a DEFINE command on a reserved object name, using an incorrect object type or subtype. The object is only allowed to be of the predetermined type listed in this topic:

Table 13. Mapping reserved objects to object types

Type	Object
Any Queue	SYSTEM.ADMIN.ACTIVITY.QUEUE SYSTEM.ADMIN.CHANNEL.EVENT SYSTEM.ADMIN.COMMAND.EVENT SYSTEM.ADMIN.CONFIG.EVENT SYSTEM.ADMIN.PERFM.EVENT SYSTEM.ADMIN.QMGR.EVENT SYSTEM.ADMIN.PUBSUB.EVENT SYSTEM.ADMIN.TRACE.ROUTE.QUEUE
Alias queue	SYSTEM.DEFAULT.ALIAS.QUEUE
Alias or local queue	SYSTEM.ADMIN.COMMAND.QUEUE SYSTEM.COMMAND.INPUT
Local queue	SYSTEM.CHANNEL.INITQ SYSTEM.CHANNEL.SYNCQ SYSTEM.CHLAUTH.DATA.QUEUE SYSTEM.CLUSTER.COMMAND.QUEUE SYSTEM.CLUSTER.REPOSITORY.QUEUE SYSTEM.CLUSTER.TRANSMIT.QUEUE SYSTEM.DEFAULT.LOCAL.QUEUE SYSTEM.QSG.CHANNEL.SYNCQ SYSTEM.QSG.TRANSMIT.QUEUE

Table 13. Mapping reserved objects to object types (continued)	
Type	**Object**
Model queue	SYSTEM.COMMAND.REPLY.MODEL SYSTEM.DEFAULT.MODEL.QUEUE SYSTEM.JMS.TEMPQ.MODEL SYSTEM.MQEXPLORER.REPLY.MODEL
Remote queue	SYSTEM.DEFAULT.REMOTE.QUEUE
Cluster-sender channel	SYSTEM.DEF.CLUSSDR
Cluster-receiver channel	SYSTEM.DEF.CLUSRCVR
Sender channel	SYSTEM.DEF.SENDER
Server channel	SYSTEM.DEF.SERVER
Receiver channel	SYSTEM.DEF.RECEIVER
Requester channel	SYSTEM.DEF.REQUESTER
Client-connection channel	SYSTEM.DEF.CLNTCONN
Server-connection channel	SYSTEM.ADMIN.SVRCONN SYSTEM.DEF.SVRCONN
Authentication information	SYSTEM.DEFAULT.AUTHINFO.CRLLDAP
Namelist	SYSTEM.DEFAULT.NAMELIST
Process	SYSTEM.DEFAULT.PROCESS
Storage class	SYSTEMST

System action

The command is ignored.

System programmer response

Ensure that reserved objects are defined with the correct object type or subtype.

CSQM109E

csect-name DYNAMIC QUEUE *value* NOT DELETED, MQRC=*mqrc (mqrc-text)*

Severity

8

Explanation

A dynamic queue could not be deleted during normal close processing, thread termination, or the end of queue manager restart, because an error occurred whilst attempting to delete it. *mqrc* gives the reason code for the error.

System action

The named dynamic queue is not deleted.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about the reason code to determine why the queue could not be deleted, and take the appropriate action as necessary. The most likely reason codes are:

- MQRC_OBJECT_IN_USE
- MQRC_PAGESET_ERROR
- MQRC_Q_NOT_EMPTY

CSQM110I

csect-name keyword(value) QSGDISP(*disposition*) HAS INCOMPLETE UNITS OF RECOVERY

Severity
8

Explanation

A command was issued that refers to a local queue that has incomplete units of recovery outstanding for it.

System action

The command is ignored.

System programmer response

Wait until all units of recovery for this queue are complete before attempting to issue the command again.

CSQM111E

csect-name COULD NOT PUT TO THE DEAD QUEUE, MQRC=*mqrc (mqrc-text)*

Severity
4

Explanation

An attempt to put a message to the dead letter queue was unsuccessful. *mqrc* gives the reason code for the error.

System action

Processing continues.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqrc* (*mqrc-text* provides the MQRC in textual form) to determine the cause of the problem.

CSQM112E

csect-name ERROR ACCESSING *keyword(value)*

Severity
4

Explanation

While processing a command for an object, object information could not be accessed. This may be because of an error on page set zero, or in the coupling facility information, or because a coupling facility structure has failed, or because Db2 is not available or is suspended. This message is issued with message CSQM090E or CSQM091E, which include a reason code that gives more information about the error.

System action

The command is ignored.

System programmer response

Check for error messages on the console log that might relate to the problem. Verify that page set zero is set up correctly; for information about this, see Page sets. If a queue sharing group is in use, check whether the coupling facility structure has failed and check that Db2 is available and not suspended. If the accompanying message is CSQM091E, an explanation of the *mqrc* in that message, and what action to take, is given in "API completion and reason codes" on page 1151.

CSQM113E

csect-name NO SPACE FOR *keyword(value)* QSGDISP(*disposition*)

Severity
8

Explanation

A command failed because page set zero is full, or because the application structure is full, or because no more application structures are available in the coupling facility (the limit is 63).

System action

The command is not actioned.

System programmer response

Do one of the following, depending on the cause of the error:

- Increase the size of page set zero or the application structure. Refer to Managing page sets for information about how to do this.
- Reduce the number of application structures you are using.

CSQM114E

csect-name keyword(value) EXCEEDED LOCAL QUEUE LIMIT

Severity
8

Explanation

A command failed because no more local queues could be defined. There is an implementation limit of 524 287 for the total number of local queues that can exist. For shared queues, there is a limit of 512 queues in a single coupling facility structure.

System action

The command is not actioned.

System programmer response

Delete any existing queues that are no longer required.

CSQM115I

csect-name keyword(value) IS CURRENTLY IN USE, ALTER WITH FORCE NEEDED

Severity
8

Explanation

The object specified is in use. This could be because:

- It is open through the API.
- When altering the USAGE attribute of a local queue, there are messages currently on the queue.
- When altering the default transmission queue, the old queue is currently being used as a transmission queue by default.

System action

The command is ignored.

System programmer response

Either:

- Wait until the object has been closed or deleted.

  Note: MCAs for receiver channels, or the intra-group queuing (IGQ) agent, can keep the destination queues open for a while even when messages are not being transmitted, and so such queues might appear to be in use.
- Wait until the queue is emptied.

- Wait until use of the queue as a default transmission queue has ended.
- Use the ALTER command with the FORCE option.

  Note: Any subsequent API calls referencing the object will fail with a reason code of MQRC_OBJECT_CHANGED.

For more information about the command, see MQSC commands.

CSQM117E

csect-name ERROR ACCESSING *keyword(value)* QSGDISP(*disposition*)

Severity
4

Explanation

While processing a command for an object, object information could not be accessed. This may be because of an error on page set zero, or in the coupling facility information, or because a coupling facility structure has failed, or because Db2 is not available or is suspended. This message is issued with message CSQM090E or CSQM091E, which include a reason code that gives more information about the error.

System action

The command is ignored.

System programmer response

Check for error messages on the console log that might relate to the problem. If *disposition* is QMGR, COPY, or PRIVATE, verify that page set zero is set up correctly; for information about this, see Page sets. If *disposition* is GROUP or SHARED, check whether the coupling facility structure has failed and check that Db2 is available and is not suspended. If the accompanying message is CSQM091E, see “API completion and reason codes” on page 1151 for an explanation of the *mqrc* in that message, and what action to take.

CSQM118I
csect-name keyword(value) QSGDISP(*disposition*) LEVEL IS INCOMPATIBLE

Explanation

The definition level of the specified object is incompatible with that of the queue manager or other members of the queue sharing group.

System action

Processing of the command is terminated.

System programmer response

For information about migration and compatibility between releases, see Maintaining and migrating.

CSQM119I
csect-name keyword(value) LEVEL IS INCOMPATIBLE

Explanation

The definition level of the specified object is incompatible with that of the queue manager or other members of the queue sharing group.

System action

Processing of the command is terminated.

System programmer response

For information about migration and compatibility between releases, see Maintaining and migrating.

CSQM120I

csect-name keyword(value) NOT ALLOWED FOR SHARED QUEUE

Severity
8

Explanation

The specified value for the object name or attribute is not allowed for a local queue with a disposition that is shared or a model queue used to create a dynamic queue that is shared.

System action

Processing of the command is terminated.

System programmer response

Reissue the command correctly.

CSQM121I

csect-name keyword(value) NOT ALLOWED, NOT IN QUEUE SHARING GROUP

Severity
8

Explanation

The specified value for the attribute requires a queue sharing group, but the queue manager is not in a group.

System action

Processing of the command is terminated.

System programmer response

Reissue the command correctly.

CSQM122I

csect-name '*verb-name object*' COMPLETED FOR QSGDISP(*disposition*)

Severity
0

Explanation

Processing for the specified command that refers to an object with the indicated disposition has completed successfully.

System action

A command is generated specifying CMDSCOPE(*) to perform further processing on all queue managers in the queue sharing group. For example, if *disposition* is GROUP, the corresponding processing must be performed for local copies of the group object.

CSQM123I

csect-name '*keyword*' VALUE CANNOT BE CHANGED

Severity
8

Explanation

The value for the specified attribute cannot be changed.

System action

Processing of the command is terminated.

System programmer response

To change the attribute, the object must be deleted and then redefined with the new value.

CSQM124I

csect-name keyword(value) ALLOWED ONLY WITH QSGDISP(*disposition*)

Severity
8

Explanation

The specified value for the attribute is allowed only for an object that has the indicated disposition.

System action

Processing of the command is terminated.

System programmer response

Reissue the command correctly.

CSQM125I

csect-name keyword(value) QSGDISP(*disposition*) WAS NOT FOUND

Severity
8

Explanation

A command was issued that refers to an object that does not exist. That is, no object could be found with the specified name and type (and subtype, for queues and channels) and disposition in the queue sharing group.

System action

The command is ignored.

System programmer response

Check that you specified the correct name for the object, and the correct subtype (for queues and channels) or channel definition table (for deleting channels). If *disposition* is GROUP or SHARED, check that Db2 is available and is not suspended. Define the object if necessary.

Note:

1. An object of the same name and type, but of a different disposition, might already exist.
2. If you are dealing with a queue or channel object, an object of the same name, but of a different subtype, might already exist.
3. Remember that the object might have recently been deleted by someone else, or from another queue manager in the queue sharing group.

CSQM126I

csect-name '*keyword*' ONLY APPLICABLE TO LU62 PROTOCOL

Severity
8

Explanation

The named keyword can only be specified when TRPTYPE(LU62) is specified.

System action

Processing of the command is terminated.

System programmer response

Reissue the command without the named keyword.

CSQM127I

csect-name keyword(value) IS EMPTY OR WRONG TYPE

Severity
8

Explanation

A namelist used to specify a list of clusters has no names in it or does not have type CLUSTER or NONE.

System action

Processing of the command is terminated.

System programmer response

Reissue the command specifying a namelist that is not empty and has type CLUSTER or NONE.

CSQM128E

csect-name MQPUT FAILED FOR QUEUE *q-name*, MQRC=*mqrc (mqrc-text)*

Severity
8

Explanation

During the processing of a command, an attempt to put a message to the named queue failed for the specified reason.

System action

In general, the command is not actioned. If the command was REFRESH QMGR for configuration events, it might be partially completed as indicated by the preceding CSQM169I messages.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqrc* (*mqrc-text* provides the MQRC in textual form). If *mqrc* is 2003, the message could not be committed.

CSQM129I

csect-name keyword(value) HAS WRONG CHANNEL TYPE

Severity
8

Explanation

The command (or the command with the particular disposition) cannot be used with the named channel because it cannot be used for channels of that type.

System action

The command is not actioned.

System programmer response

Check that the correct channel name and disposition was specified on the command. For more information about the command, see MQSC commands.

CSQM130I

csect-name CLUSTER REQUEST QUEUED

Severity
0

Explanation

Initial processing for a command completed successfully. The command requires further action by the cluster repository manager, for which a request was queued.

This message is followed by message CSQ9022I to indicate that the command has completed successfully, in that a request has been sent. It does **not** indicate that the cluster request has

completed successfully. Such requests are processed asynchronously by the cluster repository manager; any errors are reported to the z/OS console, not to the command issuer.

System action

A request was queued for the cluster repository manager, which will process it asynchronously.

CSQM131I

csect-name CHANNEL INITIATOR NOT ACTIVE, CLUSTER AND CHANNEL COMMANDS INHIBITED

Severity
8

Explanation

A command was issued that required the channel initiator to be started.

System action

The command is not actioned.

System programmer response

Issue the START CHINIT command to start the channel initiator, and reissue the command.

CSQM132I

csect-name CHANNEL INITIATOR ALREADY ACTIVE

Severity
8

Explanation

The START CHINIT command was issued but the channel initiator is already active.

System action

The command is not actioned.

CSQM133I

csect-name UNABLE TO START CHANNEL INITIATOR

Severity
8

Explanation

A START CHINIT command was issued but the channel initiator could not be started.

This could be for one of the following reasons:

- The system did not allow the channel initiator address space to be created at this time due to a heavy system workload
- There was not enough storage to start the channel initiator address space
- The system tried to obtain more address spaces than the maximum number supported
- The queue manager was quiescing or shutting down.

System action

The command is not actioned.

System programmer response

Reissue the command when the system workload is reduced and when the queue manager is not shutting down.

CSQM134I

csect-name command keyword(value) COMMAND ACCEPTED

Severity
0

Explanation

Initial processing for a command has completed successfully. The command requires further action by the channel initiator, for which a request has been queued. Messages reporting the success or otherwise of the action will be sent to the command issuer subsequently.

System action

A request was queued for the channel initiator. Further messages will be produced when the command has been completed.

CSQM135I

csect-name NO CHANNEL INITIATOR AVAILABLE

Severity
8

Explanation

A command was issued for a shared channel, but there was no suitable channel initiator available for any active queue manager in the queue sharing group. This could be because:

- no channel initiators are running
- the channel initiators that are running are too busy to allow any channel, or a channel of the particular type, to be started

System action

The command is not actioned.

System programmer response

Start a new channel initiator (on an active queue manager where there is no channel initiator running), or try again when there are fewer channels running.

CSQM136I
COMMAND NOT ALLOWED, COMMAND SERVER UNAVAILABLE

Explanation

A command for the channel initiator was entered, but the command server is not running and not enabled so the command cannot be processed.

System action

The command is not actioned.

System programmer response

Use the START CMDSERV command to start the command server, and reissue the command.

CSQM137I

csect-name command keyword COMMAND ACCEPTED

Severity
0

Explanation

Initial processing for a command has completed successfully. The command requires further action by the channel initiator, for which a request has been queued. Messages reporting the success or otherwise of the action will be sent to the command issuer subsequently.

System action

A request was queued for the channel initiator. Further messages will be produced when the command has been completed.

CSQM138I

csect-name CHANNEL INITIATOR STARTING

Severity
0

Explanation

A START CHINIT command was issued and the channel initiator address space has been started successfully.

System action

Further messages will be produced when the channel initiator itself has started.

CSQM139I

csect-name INDXTYPE(MSGTOKEN) NOT ALLOWED FOR TEMPORARY DYNAMIC QUEUE

Severity
8

Explanation

An attempt was made to define or alter a temporary-dynamic queue from which messages could be retrieved using message tokens. This combination is not allowed.

System action

Processing of the command is terminated.

System programmer response

Reissue the command with correct values.

CSQM140I

csect-name '*keyword*' NOT ALLOWED WITH TRPTYPE(*value*)

Severity
8

Explanation

The named keyword cannot be used on a START LISTENER command for the transport type shown.

System action

The command is not actioned.

System programmer response

Reissue the command with the correct keywords.

CSQM141I

csect-name 'LUNAME' REQUIRED WITH TRPTYPE(LU62)

Severity
8

Explanation

A START LISTENER command was issued specifying TRPTYPE(LU62) but without the LUNAME keyword. The LUNAME keyword is required with TRPTYPE(LU62).

System action

The command is not actioned.

System programmer response

Reissue the command with the correct keywords.

CSQM142I

csect-name CLUSTER(*cluster-name*) REPOSITORY IS NOT ON THIS QUEUE MANAGER

Severity
8

Explanation

A RESET CLUSTER command was issued, but the queue manager does not provide a full repository management service for the specified cluster. That is, the REPOS attribute of the queue manager is not *cluster_name*, or the namelist specified by the REPOSNL attribute of the queue manager does not contain *cluster_name* or is not of type CLUSTER or NONE.

System action

Processing of the command is terminated.

System programmer response

Reissue the command with the correct values or on the correct queue manager.

CSQM143I

csect-name CLUSTER TOPICS INHIBITED DUE TO PSCLUS(DISABLED)

Severity
8

Explanation

An attempt was made to define a cluster topic when the PSCLUS queue manager attribute is set to DISABLED.

System action

Processing of the command is terminated.

System programmer response

To enable publish/subscribe clustering, alter the PSCLUS attribute on all queue managers in the cluster to ENABLED.

CSQM144I

csect-name keyword(value) CANNOT BE A CLUSTER QUEUE

Severity
8

Explanation

An attempt was made to define or alter a queue to make it part of a cluster. This is not allowed if the queue is dynamic or is one of the following reserved queues:

- SYSTEM.CHANNEL.INITQ
- SYSTEM.CHANNEL.SYNCQ
- SYSTEM.CLUSTER.COMMAND.QUEUE
- SYSTEM.CLUSTER.REPOSITORY.QUEUE
- SYSTEM.COMMAND.INPUT
- SYSTEM.QSG.CHANNEL.SYNCQ
- SYSTEM.QSG.TRANSMIT.QUEUE

System action

Processing of the command is terminated.

System programmer response

Reissue the command with the correct values.

CSQM145I

csect-name '*keyword*' VALUE REQUIRED FOR SHARED QUEUE

Severity
8

Explanation

A non-blank value must be specified for the named keyword for a local queue with a disposition that is shared or a model queue used to create a dynamic queue that is shared.

System action

Processing of the command is terminated.

System programmer response

Reissue the command with a value for the keyword added.

CSQM146I

csect-name keyword(value) VALUE IS REPEATED

Severity
8

Explanation

A keyword was entered that takes a list of values, and the named value appears more than once in the list.

System action

The command is ignored.

System programmer response

Reissue the command with the parameter specified correctly. For more information about the command, see MQSC commands.

CSQM147I

csect-name '*keyword1*' AND '*keyword2*' VALUES MUST BOTH BE BLANK OR NON-BLANK

Severity
8

Explanation

An attempt was made to define or alter an object so that it had a blank value for one of the specified keywords and a non-blank value for the other. Both of those values must either be blank or non-blank.

System action

Processing of the command is terminated.

System programmer response

Reissue the command with correct values.

CSQM148I

csect-name '*keyword*' NOT ALLOWED WITH TYPE '*value*'

Severity
8

Explanation

The named keyword cannot be specified for queues or channels of the type shown.

System action

Processing of the command is terminated.

System programmer response

Reissue the command without the named keyword.

CSQM149I

csect-name '*keyword*' REQUIRED WITH TYPE '*value*'

Severity
8

Explanation

The named keyword was not specified but is required for queues or channels of the type shown.

System action

Processing of the command is terminated.

System programmer response

Reissue the command with the named keyword added.

CSQM150I

csect-name '*keyword1*' AND '*keyword2*' VALUES ARE INCOMPATIBLE

Severity
8

Explanation

An attempt was made to define or alter an object so that it had incompatible values for the specified keywords.

System action

Processing of the command is terminated.

System programmer response

Reissue the command with correct values. For information about the restrictions on the values for the keywords, see MQSC commands.

CSQM151I

csect-name '*keyword1*' AND '*keyword2*' VALUES CANNOT BOTH BE NON-BLANK

Severity
8

Explanation

An attempt was made to define or alter an object so that it had non-blank values for both of the specified keywords. At most one of those values can be non-blank.

System action

Processing of the command is terminated.

System programmer response

Reissue the command with correct values.

CSQM152I

csect-name USAGE(XMITQ) NOT ALLOWED FOR CLUSTER QUEUE

Severity
8

Explanation

An attempt was made to define or alter a queue so that it was both a transmission queue and in a cluster. This is not allowed.

System action

Processing of the command is terminated.

System programmer response

Reissue the command with correct values.

CSQM153E

csect-name Db2 NOT AVAILABLE

Severity
8

Explanation

Because Db2 is not available or no longer available, the queue manager cannot handle the command for a CF structure or shared channel.

System action

Processing of the command is terminated.

System programmer response

Use the preceding messages on the z/OS console to investigate why Db2 is not available, and resume the connection or restart Db2 if necessary.

CSQM154E

csect-name ERROR ACCESSING Db2

Severity
8

Explanation

Because there was an error in accessing Db2, the queue manager cannot handle the command for a CF structure or shared channel.

System action

Processing of the command is terminated.

System programmer response

Resolve the error reported in the preceding messages.

CSQM155I

csect-name STATUS(STOPPED) NOT ALLOWED WITH QMNAME OR CONNAME

Severity
8

Explanation

An attempt was made to stop a channel using STATUS(STOPPED), but a queue manager name or connection name was also specified. This is not allowed.

System action

Processing of the command is terminated.

System programmer response

Reissue the command with correct values.

CSQM156I

csect-name INDXTYPE(GROUPID) NOT ALLOWED FOR *keyword(value)*

Severity
8

Explanation

An attempt was made to define or alter a queue with a reserved name so that it had an index type of GROUPID. This is not allowed.

System action

Processing of the command is terminated.

System programmer response

Reissue the command with correct values.

CSQM157E

csect-name NO SPACE FOR *keyword(value)*

Severity
8

Explanation

An IBM MQ DEFINE CFSTRUCT command failed because no more application structures are available in the coupling facility (the limit is 63).

System action

The command is not actioned.

System programmer response

Reduce the number of application structures you are using.

CSQM158I

csect-name RECOVER(YES) NOT ALLOWED WITH CFLEVEL(*value*)

Severity
8

Explanation

An attempt was made to define or alter a CF structure to support recovery, but the level of the CF structure was less than 3. This is not allowed.

System action

Processing of the command is terminated.

System programmer response

Reissue the command with correct values. You cannot alter the level of a CF structure; you must delete the structure and then redefine it.

CSQM159I

csect-name verb-name object(obj-name) NOT ALLOWED, INCOMPATIBLE QUEUE MANAGER CMDLEVELS

Severity
8

Explanation

An attempt was made to alter the CF level of a CF structure, or to delete the structure. This action requires that all queue managers in the queue sharing group must have a certain command level. Some of the queue managers have a lower level.

System action

Processing of the command is terminated.

System programmer response

Ensure all the queue managers in the queue sharing group have the appropriate command level. For information about restrictions on the command, see MQSC commands.

CSQM160I

csect-name keyword(value) IS NOT UNIQUE

Severity
8

Explanation

A command was issued that refers to an object that exists with more that one disposition in the queue sharing group, so the object to be used cannot be determined.

System action

The command is not executed.

System programmer response

Delete one of the objects.

CSQM161I

csect-name QUEUE ATTRIBUTES ARE INCOMPATIBLE

Severity
8

Explanation

A MOVE QLOCAL command was issued, but the queues involved have different values for one or more of these attributes: DEFTYPE, HARDENBO, INDXTYPE, USAGE. Messages cannot be moved safely if these attributes differ.

System action

The command is not executed.

System programmer response

Check that the queue names have been entered correctly. Change the queue attributes as necessary.

CSQM162I

csect-name keyword(value) MAXDEPTH IS TOO SMALL

Severity
8

Explanation

A MOVE QLOCAL command was issued, but the MAXDEPTH attribute value for the target queue is too small to allow all the messages to be moved.

System action

The command is not executed.

System programmer response

Change the MAXDEPTH value for the queue.

CSQM163I

csect-name ERROR USING *keyword(value)*, MQRC=*mqrc (mqrc-text)*

Severity
8

Explanation

During the processing of a MOVE QLOCAL command, an attempt to open the named queue or to get or put a message for it failed for the specified reason. For example, a put to the target queue will fail if a message is too long.

System action

The command stops processing. If some messages have already been moved and committed, they will remain on the target queue; the rest of the messages will not be moved.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqrc* (*mqrc-text* provides the MQRC in textual form), and take the appropriate action to resolve the problem.

CSQM164I

csect-name keyword(value) HAS MESSAGES ASSOCIATED WITH IT

Severity
8

Explanation

A MOVE QLOCAL command was issued specifying TYPE(MOVE), the target queue already has messages associated with it.

System action

The command is not executed.

System programmer response

Check that the queue name was entered correctly. Determine if it is safe to add messages to the queue, then reissue the command using the TYPE(ADD) option.

CSQM165I

csect-name n MESSAGES MOVED

Severity
0

Explanation

A MOVE QLOCAL command was issued, and moved the indicated number of messages.

If the command completed successfully and moved all the messages on the queue, this confirms the number moved. If an error occurred while moving the messages, this shows how many messages were successfully moved to the target queue and committed.

System action

Processing continues.

System programmer response

If the command did not complete successfully, as shown by the following CSQ9023E message, investigate the problem reported in the preceding messages.

CSQM166I

csect-name keyword(value) NOT AUTHORIZED

Severity
8

Explanation

You do not have proper authorization to use the command for the specified object.

System action

The command is not executed for that object.

System programmer response

Check that the object name was entered correctly. If required, arrange for someone who is authorized to use the object to issue the command for you, or get the necessary authority granted to you.

CSQM167I

csect-name PERFORMANCE EVENTS DISABLED

Severity
8

Explanation

A command was issued that required performance events to be enabled.

System action

The command is not executed.

System programmer response

Use the ALTER QMGR command to set the PERFMEV attribute to ENABLED if performance events are required.

CSQM168I

csect-name CONFIGURATION EVENTS DISABLED

Severity
8

Explanation

A command was issued that required configuration events to be enabled.

System action

The command is not executed.

System programmer response

Use the ALTER QMGR command to set the CONFIGEV attribute to ENABLED if configuration events are required.

CSQM169I

csect-name object-type OBJECTS: *m* FOUND, *n* EVENTS GENERATED

Severity
0

Explanation

A REFRESH QMGR command was issued for configuration events. *m* objects of the indicated type were found that matched the specified selection criteria (such as name or time of alteration), and *n* event messages were generated. The number of event messages might be less than the number of objects found because certain objects might be excluded, such as temporary dynamic queues or objects in the process of being deleted. It might also be less than the number of objects found if there was a problem with the event queue.

System action

Processing continues.

System programmer response

If *n* is less than *m*, but message CSQ9022I follows these messages to indicate that the command completed successfully, no action is needed. Otherwise, investigate the problem with the event queue as reported in the preceding messages.

CSQM170I

csect-name REFRESHING CONFIGURATION EVENTS SINCE *date time*

Severity
0

Explanation

A REFRESH QMGR command was issued for configuration events specifying a refresh interval with the INCLINT keyword. Event messages will be generated for all objects with an alteration date and time later than *date time* (provided they match any other specified selection criteria, such as name or type). However, event messages will not be generated for objects deleted after that time.

CSQM171I

csect-name CONFIGURATION EVENTS REFRESH NEEDED

Severity
0

Explanation

An ALTER QMGR command was issued that enables configuration events. Event messages need to be generated to ensure that the configuration information is complete and up to date.

System action

Processing continues.

System programmer response

If complete configuration information is required, do one of the following, as appropriate:

- If this is the first time that configuration events have been enabled, use the REFRESH QMGR TYPE(CONFIGEV) command to generate configuration events for **all** objects. If you have many objects, it may be preferable to use several such commands each with a different selection of objects, but such that all are included.
- Otherwise, use the REFRESH QMGR TYPE(CONFIGEV) command to generate events to replace those that were not generated while configuration events were disabled; specify the INCLINT parameter to cover this period.

CSQM172I

csect-name '*keyword*' NOT ALLOWED WITH TYPE(*value*)

Severity
8

Explanation

The named keyword cannot be specified with the TYPE value shown.

System action

Processing of the command is terminated.

System programmer response

Reissue the command without the named keyword.

CSQM173I

csect-name EXPIRED MESSAGE SCAN REQUESTED FOR *m* QUEUES

Severity
0

Explanation

A REFRESH QMGR command was issued for expired message scanning. *m* queues were found that matched the specified selection criteria.

System action

Processing continues.

CSQM174E

csect-name '*keyword*' is not allowed with CFLEVEL(*cflevel*) - this keyword requires CFLEVEL(5)

Severity
8

Explanation

An attempt was made to define or alter the value of a structure attribute related to SMDS, but the level of the structure was less than CFLEVEL(5). This is not allowed.

System action

Processing for the command is terminated.

System programmer response

Issue the command again with correct values. You cannot alter the level of a CF structure; you must delete the structure, and then define it again.

CSQM175E

csect-name '*keyword*' cannot be altered because a data set is currently active for this structure

Severity
8

Explanation

The keywords DSGROUP and DSBLOCK can only be altered before the first data set has been allocated for the structure. Once an SMDS data set has become active for this structure then these attribute values cannot be changed.

System action

Processing for the command is terminated.

System programmer response

Verify the command entry and reissue the command correctly.

CSQM176E

csect-name SMDS cannot currently be reset to *keyword*(*value*)

Severity
8

Explanation

A **RESET SMDS** command requested a change of status which is not compatible with the existing status.

- The option **STATUS(FAILED)** is only allowed when the current status is **ACTIVE** or **RECOVERED** (or already **FAILED**, in which case the command has no effect).
- The option **STATUS(RECOVERED)** is only allowed when the current status is **FAILED** (or already **RECOVERED**).

System action

Processing for the command is terminated.

System programmer response

Verify the command entry, and reissue the command correctly.

CSQM177I

csect-name '*keyword*' NOT ALLOWED WITH ACTION '*value*'

Severity
8

Explanation

The named keyword cannot be specified for channel authentication settings of the action shown.

System action

Processing for the command is terminated.

System programmer response

Reissue the command without the named keyword.

CSQM178I

csect-name ACTION NOT ALLOWED FOR CHANNEL *channel-type*(*channel-name*)

Severity
8

Explanation

The MATCH(RUNCHECK) action that you requested cannot be performed on the channel with the specified parameters. This may be because either: -

- The channel is a SVRCONN and the QMNAME parameter was supplied.
- The channel is not a SVRCONN and the CLNTUSER parameter was supplied

System action

Processing of the command is terminated.

System programmer response

Either correct the specified parameters or alter the channel to the appropriate channel type and then reissue the command.

CSQM179I

csect-name CHANNEL WILL RUN USING MCAUSER(*userid*)

Severity
0

Explanation

No matching channel authentication (CHLAUTH) records were found that match the given fields.

Note:

1. The returned MCAUSER value does not take into account possible actions by a channel security exit.
2. Channel authentication rules that match the host name apply only if the queue manager REVDNS attribute is enabled and the DNS server returns a valid host name for the IP address.

CSQM181I

csect-name INSUFFICIENT STORAGE TO COMPLETE COMMAND

Severity
8

Explanation

There was insufficient storage available to complete processing for the command.

System action

The command terminates. Any processing already completed may be retained or backed out.

System programmer response

Refer to the accompanying messages to determine what processing has been done. Retry the command, if appropriate, when your queue manager is less busy. If the problem persists, you might need to increase the region size used by your queue manager, or you might need to reduce the number of jobs running in your system.

CSQM182E

csect-name DURABLE SUBSCRIPTIONS NOT ALLOWED

Severity
8

Explanation

A DEFINE SUB command was issued, but it was not possible to make a durable subscription.

This could be for one of the following reasons:

- The topic subscribed to is defined as DURSUB(NO)
- The queue named SYSTEM.DURABLE.SUBSCRIBER.QUEUE is not available
- The CSQINP2 data sets are in the wrong order, the order is:

```
 //CSQINP2  DD DSN=hlq.SCSQPROC(CSQ4INYS),DISP=SHR //         DD
DSN=hlq.SCSQPROC(CSQ4INSX),DISP=SHR
 //         DD DSN=hlq.SCSQPROC(CSQ4INSG),DISP=SHR
```

System action

The command is not executed.

System programmer response

Durable subscriptions are stored on the SYSTEM.DURABLE.SUBSCRIBER.QUEUE. Ensure that this queue is available for use. Possible reasons for failure include the queue being full, the queue being put inhibited, or the queue not existing.

If the topic subscribed to is defined as DURSUB(NO) then it is not possible to administratively define a subscription. The topic can be altered to DURSUB(YES) to enable the subscription to be defined.

CSQM183E

csect-name SUBSCRIPTION INHIBITED

Severity
8

Explanation

A DEFINE SUB command was issued, but it was not possible to make a subscription because the topic subscribed to is defined as SUB(DISABLED).

System action

The command is not executed.

System programmer response

If the topic subscribed to is defined as SUB(DISABLED) then it is not possible to administratively define a subscription. The topic can be altered to SUB(ENABLED) to enable the subscription to be defined.

CSQM184I

csect-name '*keyword1*' AND '*keyword2*' VALUES CANNOT BOTH BE BLANK

Severity
8

Explanation

An attempt was made to define or alter an object so that it had blank values for both of the specified keywords. One of those values must be provided.

System action

Processing of the command is terminated.

System programmer response

Reissue the command with correct values.

CSQM185E

csect-name SUBSCRIPTION HAS FIXED SUBUSER

Severity
8

Explanation

An ALTER SUB command was issued, but it was not possible to ALTER the target subscription because the userid performing the ALTER did not match the SUBUSER attribute of the subscription and the subscription has had the VARUSER(FIXED) attribute set.

System action

The command is not executed.

System programmer response

The subscription can be altered only by the owning userid that is displayed in the SUBUSER attribute.

CSQM186E

csect-name DESTCLAS VALUE CANNOT BE ALTERED

Severity
8

Explanation

An ALTER SUB command was issued, but it was not possible to ALTER the target subscription because the DESTCLAS attribute specified on the request did not match the one in the existing subscription. DESTCLAS cannot be altered.

System action

The command is not executed.

System programmer response

Ensure that the DESTCLAS attribute matches the existing subscription and rerun the request.

CSQM187E

csect-name GROUPING VALUE CANNOT BE ALTERED

Severity
8

Explanation

An ALTER SUB command was issued, but it was not possible to ALTER the target subscription because the GROUPING attribute specified on the request did not match the one in the existing subscription. GROUPING attributes cannot be altered.

System action

The command is not executed.

System programmer response

Ensure that the GROUPING attribute matches the existing subscription and rerun the request.

CSQM188E

csect-name SUBSCOPE VALUE CANNOT BE ALTERED

Severity
8

Explanation

An ALTER SUB command was issued, but it was not possible to ALTER the target subscription because the SUBSCOPE attribute specified on the request did not match the one in the existing subscription. SUBSCOPE cannot be altered.

System action

The command is not executed.

System programmer response

Ensure that the SUBSCOPE attribute matches the existing subscription and rerun the request.

CSQM189E

csect-name SELECTOR VALUE CANNOT BE ALTERED

Severity
8

Explanation

An ALTER SUB command was issued, but it was not possible to ALTER the target subscription because the SELECTOR attribute specified on the request did not match the one in the existing subscription. SELECTOR cannot be altered.

System action

The command is not executed.

System programmer response

Ensure that the SELECTOR attribute matches the existing subscription and rerun the request.

CSQM190E

csect-name TOPIC STRING IS INVALID

Severity
8

Explanation

A DEFINE SUB command was issued, but it was not possible to make a subscription because the topic string was invalid.

This could be because the WSCHEMA attribute was set to CHAR and either:

- The TOPICSTR attribute contains an invalid escape character, or
- The TOPICOBJ attribute refers to a TOPIC object with a TOPICSTR attribute that contains an invalid escape character.

System action

The command is not executed.

System programmer response

Correct the TOPICSTR attribute on the **DEFINE SUB** command to correctly use escape characters. If the problem is with the TOPICSTR in a TOPIC object, correct that TOPIC object or refer to a different TOPIC object. If the TOPICSTR needs to use the characters in that way, set the WSCHEMA attribute to *TOPIC* to avoid errors with escape characters.

CSQM191E

csect-name TOPIC STRING CANNOT BE ALTERED

Severity
8

Explanation

A DEFINE TOPIC command using the REPLACE keyword was issued, providing a value for TOPICSTR that was different from the value in the existing object. This is not allowed.

System action

The command is not executed.

System programmer response

Reissue the command with correct values. You cannot alter the topic string in a topic object; you must delete the object and then redefine it.

CSQM192I

csect-name Address '*address*' is invalid.

Severity
8

Explanation

The IP address or host name *address* contains invalid characters.

System action

Processing of the command is terminated.

System programmer response

Reissue the command with the parameter specified correctly. Note that the BLOCKADDR list may contain only IP addresses: host name addresses are not permitted.

CSQM193I

csect-name IP address '*ipaddress*' contains an invalid range.

Severity
8

Explanation

The IP address *ipaddress* contains an invalid range. For example, the lower number is greater than or equal to the upper number for the range.

System action

Processing of the command is terminated.

System programmer response

Reissue the command with the parameter specified correctly.

CSQM194I

csect-name IP address '*ipaddress1*' overlaps existing IP address '*ipaddress2*'.

Severity
8

Explanation

The IP address *ipaddress1* overlaps with an existing IP address *ipaddress2*. For example, addresses 1.2.3.4-7 and 1.2.3.6-8 overlap.

System action

Processing of the command is terminated.

System programmer response

Reissue the command with the parameter specified correctly.

CSQM195I

csect-name MATCH RUNCHECK FOUND A GENERIC VALUE IN *field-name*

Severity
8

Explanation

A DISPLAY **CHLAUTH** command was issued using the MATCH(RUNCHECK) parameter and the *field-name* parameter was found to contain a generic value, which is not allowed.

System action

Processing of the command is terminated.

System programmer response

Reissue the command with a value in *field-name* which is not generic.

CSQM196I

csect-name REQUIRED KEYWORD MISSING FOR *keyword*(*value*)

Severity
8

Explanation

A required additional keyword was not specified in conjunction with *keyword* (*value*).

This message can be returned in the following scenarios:

- A **DISPLAY CHLAUTH** command, specifying **MATCH(RUNCHECK)** did not specify the **ADDRESS** keyword or one of the keywords **CLNTUSR** or **QMNAME**.
- A **SET CHLAUTH** command, the **MCAUSER** is missing when **USERSRC(MAP)** is specified or **USERSRC** is missing as **USERSRC(MAP)** is the default.

System action

Processing of the command is terminated.

System programmer response

Reissue the command specifying one of the required keywords

CSQM197I

csect-name '*keyword*' NOT ALLOWED WITH MATCH '*value*'

Severity
8

Explanation

The named keyword cannot be specified for **DISPLAY CHLAUTH** in conjunction with the identified value for the **MATCH** keyword.

System action

Processing of the command is terminated.

System programmer response

Reissue the command without the named keyword.

CSQM198I

csect-name CHANNEL AUTHENTICATION PROFILE NAME IS INVALID

Severity
8

Explanation

The channel profile name used in the command was not valid.

System action

Processing of the command is terminated.

System programmer response

Check that the characters entered for the profile are valid and reissue the command. If TYPE(BLOCKADDR) is specified, check that CHLAUTH('*') is also specified

CSQM199I

csect-name CFCONLOS (TOLERATE) NOT ALLOWED, INCOMPATIBLE QUEUE MANAGER CMDLEVELS

Severity
8

Explanation

An attempt was made to change the **CFCONLOS** queue manager attribute to a value of **TOLERATE**, which enables toleration of loss of connectivity to Coupling Facility structures. This action requires that all queue managers in the queue sharing group must have a command level of at least 710. Some of the queue managers have a lower level.

System action

Processing of the command is terminated.

System programmer response

Ensure all the queue managers in the queue sharing group have the appropriate command level. For information about restrictions on the command, see MQSC commands.

V 10.0.0 **CSQM200I**
csect-name TRPTYPE(LU62) NOT ALLOWED WITH A CONNAME >48

Severity
8

Explanation
An attempt was made to specify a **CONNAME** value of greater than 48 characters for a channel with a transmission protocol type of LU6.2. A **CONNAME** value of greater than 48 characters can only be specified for a channel with a transmission protocol type of TCP.

System action
Processing of the command is terminated.

System programmer response

Specify a **CONNAME** value of up to 48 characters and reissue the command. For more information about commands, see MQSC commands.

CSQM201I

csect-name verb-name *obj-type* DETAILS

Severity
0

Explanation

This message is the response to a command that displays attributes or other information about objects, when that command was entered from either the console, or the command server initialization server. It shows the attributes requested for *obj-type*, as follows:

```
obj-type(name)  attribute-value  attribute-value ⋮  END obj-type DETAILS
```

See the specific command for details of the attributes and values.

csect-name might include the command prefix (CPF), depending on how the command was entered.

Exceptionally, the last line might be:

```
obj-type TERMINATED WITH MAX LINES
```

if the number of lines allowed in a multiple line WTO to be issued on the console (255) was exceeded. This figure includes the first and last lines of the display. The only object that might cause this message is namelist because displaying a complete namelist would require 263 lines in total. (This only occurs when the command was issued from the console.) For details of the fields reported, see the command description.

CSQM224I

csect-name verb-name obj-type DETAILS - CURRENTLY DISABLED

Severity
0

Explanation

This message is issued instead of CSQM201I for channel authentication (CHLAUTH) records if the CHLAUTH queue manager attribute has been set to DISABLED.

See the explanation of message CSQM201I for more information.

CSQM292I

csect-name PUBLISH/SUBSCRIBE ENGINE IS DISABLED

Severity
0

Explanation

The publish/subscribe engine is unavailable because it has been disabled.

System action

The command is actioned, but no results are returned because the publish/subscribe engine has been disabled.

System programmer response

This message occurs because you are attempting to query the publish/subscribe engine but you have disabled it. To use the publish/subscribe engine, set the PSMODE queue manager attribute to a value other than DISABLED.

CSQM293I

csect-name m obj-type FOUND MATCHING REQUEST CRITERIA

Severity
0

Explanation

A command that displays attributes or other information about objects has been issued. *m* objects were found that matched the specified selection criteria.

System action

For each object found, a message follows giving its details.

CSQM294I

csect-name CANNOT GET INFORMATION FROM DB2

Severity
8

Explanation

While processing a command that displays attributes or other information about objects with a disposition of GROUP or SHARED, information could not be obtained from Db2. This might be because Db2 is not available or no longer available, or because it is suspended, or because there was an error in accessing Db2, or because a Db2 table was temporarily locked.

System action

Information about objects with a disposition of GROUP or SHARED is not displayed, so the information displayed might therefore be incomplete.

System programmer response

Refer to the console log for messages giving more information about the error.

CSQM295I

csect-name UNEXPECTED ERROR DURING DISPLAY

Severity
8

Explanation

A severe error occurred while processing a command that displays attributes or other information about objects.

System action

The command is terminated.

System programmer response

Refer to the console log for messages giving more information about the error.

CSQM297I

csect-name NO *item* FOUND MATCHING REQUEST CRITERIA

Severity
0

Explanation

A command that displays attributes or other information about objects or runtime status found that there are no items that match the specified name and satisfy any other criteria requested (such as subtype or disposition in a queue sharing group).

CSQM298I

csect-name TOTAL MESSAGE LENGTH ALLOWED ON CONSOLE EXCEEDED

Severity
8

Explanation

The total message length for the command allowed on the console (32 K) was exceeded.

System action

The command is actioned, but the display of the command is terminated.

System programmer response

This error occurs if a command that displays attributes or other information about objects is entered using a generic name (for example, DIS Q(*) ALL), and the total amount of data to be displayed exceeds 32 K. To avoid this problem, try to be more selective about the information requested (for example, DIS Q(PAY*) ALL).

CSQM299I

csect-name INSUFFICIENT STORAGE TO COMPLETE DISPLAY

Severity
8

Explanation

There was insufficient storage available to complete processing of a command that displays attributes or other information about objects.

System action

The command is actioned, but the display of the information is terminated before completion. The data returned is a subset of the requested information. Refer to message CSQM293I, which indicates how many objects have information returned. The message does not indicate how many matching objects were found.

System programmer response

If this error occurs when a generic name is used in the command (for example, DIS QUEUE(*) ALL), try to be more selective about the information requested (for example, DIS QUEUE(PAY*) ALL). If the problem persists, you might need to increase the region size used by your queue manager or channel initiator, or you might need to reduce the number of jobs running in your system.

CSQM4nnI

object details

Severity
0

Explanation

This message consists of the entire object or object status details formatted for use by applications. It is issued in response to commands entered from the command server. Message CSQ9022I follows this message.

The message number depends on the object or object status type, as follows:

Table 14. Mapping message numbers to object or status types

Number	Object or status type
CSQM400I	Storage class object
CSQM401I	Local queue object

Table 14. Mapping message numbers to object or status types (continued)	
Number	**Object or status type**
CSQM402I	Model queue object
CSQM403I	Alias queue object
CSQM406I	Remote queue object
CSQM407I	Namelist object
CSQM408I	Process object
CSQM409I	Queue manager object
CSQM410I	Sender channel object
CSQM411I	Server channel object
CSQM412I	Receiver channel object
CSQM413I	Requester channel object
CSQM415I	Server-connection channel object
CSQM416I	Client-connection channel object
CSQM417I	Cluster-receiver channel object
CSQM418I	Cluster-sender channel object
CSQM420I	Sender channel status
CSQM421I	Server channel status
CSQM422I	Receiver channel status
CSQM423I	Requester channel status
CSQM425I	Server-connection channel status
CSQM427I	Cluster-receiver channel status
CSQM428I	Cluster-sender channel status
CSQM430I	CF structure object
CSQM431I	Cluster queue object
CSQM437I	Authentication information object
CSQM438I	Topic object
CSQM439I	Cluster queue manager object
CSQM440I	CF structure status
CSQM441I	Local queue status
CSQM442I	Connection information
CSQM443I	Topic status
CSQM444I	Subscription
CSQM445I	Subscription status
CSQM446I	Publish/Subscribe status
CSQM451I	Local queue statistics

Table 14. Mapping message numbers to object or status types (continued)	
Number	**Object or status type**
CSQM452I	Shared message data set
CSQM453I	Shared message data set connection
CSQM454I	Channel authentication record

CSQM500I

csect-name GROUPUR agent starting TCB=*tcb-name*

Severity
0

Explanation

The group unit of recovery (GROUPUR) agent was started during the initialization of a queue manager that is in a queue sharing group. The agent uses TCB *tcb-name*.

The GROUPUR agent monitors the SYSTEM.QSG.UR.RESOLUTION.QUEUE to process requests from other queue managers within the QSG.

System action

Processing continues. The GROUPUR agent is started.

CSQM501I

csect-name GROUPUR agent stopping

Severity
4

Explanation

The group unit of recovery (GROUPUR) agent is stopping because of one the following reasons:

- the queue manager is stopping
- it was unable to recover from an IBM MQ API error or an abnormal ending

System action

The GROUPUR agent stops.

If the agent has stopped due to an error it will be automatically restarted.

System programmer response

If the queue manager is not stopping, investigate the cause of the error as reported in the preceding messages.

CSQM502I

csect-name processed BACKOUT request from *qmgr-name* for in-doubt UOW, URID=*urid*, CONNECTION-NAME=*name*

Severity
0

Explanation

This message is generated during queue manager startup when the GROUPUR agent has processed a message on the SYSTEM.QSG.UR.RESOLUTION.QUEUE from another queue manager in the queue sharing group requesting that the specified UOW be backed out.

System action

Processing continues.

CSQM503I

csect-name processed COMMIT request from *qmgr-name* for in-doubt UOW, URID=*urid*, CONNECTION-NAME=*name*

Severity
0

Explanation

This message is generated during queue manager startup when the GROUPUR agent has processed a message on the SYSTEM.QSG.UR.RESOLUTION.QUEUE from another queue manager in the queue sharing group requesting that the specified UOW be committed.

System action

Startup continues.

CSQM504I

csect-name GROUPUR support enabled

Severity
0

Explanation

This message is generated during queue manager startup, or in response to an ALTER QMGR command, if the GROUPUR queue manager attribute is enabled and all of the configuration checks performed by the GROUPUR agent are satisfied.

System action

The queue manager permits applications to establish transactions with a GROUP unit of recovery disposition.

CSQM505I

csect-name GROUPUR support disabled

Severity
0

Explanation

This message is generated during queue manager startup or in response to an ALTER QMGR command if the GROUPUR queue manager attribute is disabled.

System action

The queue manager inhibits applications from establishing transactions with a GROUP unit of recovery disposition.

CSQM506I

csect-name GROUPUR qmgr attribute has been disabled CODE=*code*

Severity
4

Explanation

This message is generated at queue manager startup if the GROUPUR queue manager attribute is enabled but one of the configuration checks performed by the GROUPUR agent failed. CODE=*code* contains an identifier indicating which configuration check failed.

System action

The GROUPUR queue manager attribute is disabled.

System programmer response

The system programmer should use the code specified to identify what configuration check failed. If support for group units of recovery is required, they should take corrective action and then re-enable the GROUPUR queue manager attribute.

CSQM507E

csect-name GROUPUR qmgr attribute was not enabled CODE=*code*

Severity

8

Explanation

This message is generated in response to an ALTER QMGR command if an attempt to enable the GROUPUR queue manager attribute fails because one of the configuration checks performed by the GROUPUR agent are not satisfied. CODE=*code* contains an identifier indicating which configuration check failed.

System action

The GROUPUR queue manager attribute remains disabled and the ALTER QMGR command fails.

System programmer response

The system programmer should use the code specified to identify what configuration check failed. They should then take corrective action and then re-issue the ALTER QMGR command.

When you enable group units of recovery (GROUPUR support) a number of configuration checks are performed to ensure the configuration steps have been completed. You cannot enable this support if any of these checks fail.

These checks are also performed at queue manager startup if GROUPUR queue manager attribute is enabled. If one of these checks fails during startup then group units of recovery will be disabled until you correct the error and re-enable the GROUPUR queue manager attribute.

If a check fails it will be identified with a return code (number). You can use this code to identify the failing check using the following list:

1. This queue manager is not a member of a queue sharing group.
2. The SYSTEM.QSG.UR.RESOLUTION.QUEUE does not exist.
3. The SYSTEM.QSG.UR.RESOLUTION.QUEUE does not support persistent messages.
4. The SYSTEM.QSG.UR.RESOLUTION.QUEUE is not indexed by correlation ID.
5. The SYSTEM.QSG.UR.RESOLUTION.QUEUE does not reside on the system application coupling facility structure, CSQSYSAPPL.
6. The queue manager name is the same as the name of the queue sharing group.

CSQM508E

csect-name GROUPUR agent ended abnormally. Restarting

Severity

8

Explanation

The group unit of recovery (GROUPUR) agent has ended abnormally because a severe error occurred, as reported in the preceding messages.

System action

The group unit of recovery (GROUPUR) agent attempts to restart a number of times. If it fails persistently, it terminates.

System programmer response

Ensure the CFSTRUCT called CSQSYSAPPL is configured for GROUPUR operation. See Enabling GROUP units of recovery.

Investigate the reason for the abnormal termination, as reported in the preceding messages.

CSQM520I

csect-name PSCLUS CANNOT BE ALTERED, CLUSTER TOPICS EXIST

Severity
8

Explanation

An attempt was made to set the PSCLUS queue manager attribute to DISABLED, indicating that Publish/Subscribe activity is not expected in this cluster between queue managers, but a cluster topic exists so the setting cannot be modified.

System action

Processing of the command is terminated.

System programmer response

To disable publish/subscribe clustering delete all cluster topic objects before altering the PSCLUS attribute on all queue managers in the cluster to DISABLED.

CSQM521I

csect-name CLCHNAME MUST BE BLANK FOR DYNAMIC QUEUE

Severity
8

Explanation

An attempt was made to define or alter a dynamic queue with a non blank value for the CLCHNAME attribute, which is not allowed.

System action

Processing of the command is terminated.

System programmer response

Reissue the command with compatible attribute values.

CSQM522I

csect-name NOSHARE NOT ALLOWED WITH NON-BLANK CLCHNAME

Severity
8

Explanation

An attempt was made to define or alter a queue with a non-blank value for the CLCHNAME attribute, but NOSHARE was specified or implied.

System action

Processing of the command is terminated.

System programmer response

Reissue the command, specifying either SHARE or a blank value for the CLCHNAME attribute.

CSQM523I

csect-name CLUSTER OR CLROUTE CANNOT CURRENTLY BE ALTERED

Severity
8

Explanation

An attempt was made to alter an administered topic that is currently in a named cluster. While a topic is in a cluster it is not permitted to modify the CLROUTE attribute, or to modify the CLUSTER attribute to an alternative cluster name if CLROUTE is set to TOPICHOST.

System action

Processing of the command is terminated.

System programmer response

To alter the CLROUTE or CLUSTER attribute, perform the following actions:

1. Quiesce publish/subscribe messaging for the topic.
2. Remove the topic from the cluster by setting the value of the CLUSTER attribute to blank.
3. Set the CLROUTE and CLUSTER attributes to their required value once the topic has been removed from the cluster.
4. Resume publish/subscribe messaging once the change is visible in the cluster and the queue manager has received proxy subscriptions for any remote subscriptions.

CSQM524I

csect-name CLROUTE CONFLICT DETECTED FOR CLUSTER TOPIC

Severity
8

Explanation

An attempt was made to define a cluster topic but the value of the CLROUTE attribute conflicts with an existing topic, either above or below it, in the topic tree.

System action

Processing of the command is terminated.

System programmer response

Review the cluster routing requirements for the topic tree, then correct and reissue the command.

CSQM525I

csect-name obj-type(obj-name) DOES NOT EXIST OR IS DEFINED INCORRECTLY

Severity
8

Explanation

The queue manager could not complete a requested operation because an object named *obj-name* of type *obj-type* does not exist or is defined incorrectly.

System action

Processing of the operation is terminated.

System programmer response

Check the object has been defined correctly, then try the operation again.

For information on how to define system objects, see Sample definitions supplied with IBM MQ.

If this message has been issued for the model queue SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE, and the queue has been defined, ensure it has the following attributes:

- The **USAGE** attribute must be set to **XMITQ**
- The **QSGDISP** attribute must not be **SHARED**

- The **DEFTYPE** attribute must be set to **PERMDYN**
- The **INDXTYPE** attribute must be set to **CORRELID**
- The **SHARE** attribute must be set

CSQM526I

csect-name CERTIFICATE LABEL NOT ALLOWED FOR SSLv3 CHANNEL

Severity
8

Explanation

An attempt was made to specify a certificate label for an inbound channel that uses a SSL v3 CipherSpec, which is not allowed. Certificate labels for inbound channels are only supported for TLS channels.

System action

Processing of the command is terminated.

System programmer response

If you need to configure a certificate label, alter the channel to use a TLS CipherSpec.

CSQM527E

csect-name attribute(value) NOT ALLOWED WITH CHANNEL TYPE *channel-type*

Severity
8

Explanation

An attempt was made to specify a channel attribute value combination which is not applicable for the specified channel type.

Either SPLPROT(ASPOLICY) was specified on a channel with CHLTYPE SDR or SVR, or SPLPROT(REMOVE) was specified on a channel with CHLTYPE RCVR or RQSTR. These combinations are not valid.

System action

Processing of the command is terminated.

System programmer response

Ensure a valid attribute and value combination is specified for the respective channel type. For more information, see SPLPROT under the DEFINE CHANNEL command.

CSQM528E

SSLCIPH(*cipher specification name*) is not supported

Severity
8

Explanation

The specified cipher specification name is not supported by the queue manager.

System action

An attempt to define or alter a channel definition fails.

System programmer response

Refer to CipherSpecs order in TLS handshake for a list of supported cipher specifications, and specify one of these.

CSQM529E

SSLCIPH(*cipher specification name*) is not in the AllowedCipherSpecs list

Severity
8

Explanation

The specified cipher specification name is not in the list of AllowedCipherSpecs defined in the QMINI data set specified in the queue manager start up procedure.

System action

An attempt to define or alter a channel definition fails.

System programmer response

Define a cipher specification that is in the list of AllowedCipherSpecs, or add the cipher specification to AllowedCipherSpecs and restart the queue manager.

CSQM530E

Unsupported cipher specification in AllowedCipherSpecs list.

Severity
8

Explanation

The AllowedCipherSpecs list in the QMINI data set names a cipher specification that is not supported by the queue manager. It is possible that the named cipher specification is a TLS V1.3 cipher specification and the version of z/OS that the queue manager is running on does not support TLS V1.3 function.

To be able to use TLS V1.3 function, the z/OS operating system must be at V2.4 or higher.

System action

An attempt to define or alter a channel definition fails.

System programmer response

Stop the queue manager. Remove or comment out TLS V1.3 cipher specifications from the AllowedCipherSpecs list in the QMINI data set, or upgrade to a version of the z/OS operating system that supports TLS V1.3. Restart the queue manager and redefine the channel with a supported cipher specification.

CSQM531I

Cannot set CAPEXPRY as both a first-class and custom attribute.

Severity
0

Explanation

A command was issued that would result in both the first-class CAPEXPRY and CUSTOM CAPEXPRY being populated with non-default values. This is not allowed.

System action

The command is ignored.

System programmer response

If this is a new object, adjust the command to define only one of the two **CAPEXPRY** parameters.

If the object is already defined, check to see which **CAPEXPRY** value is set to a non-default value. Then, adjust the command to provide non-conflicting values.

If you are attempting to set **CAPEXPRY**, ensure the **CUSTOM** value is not set, or is set to the default. If you are attempting to set the **CUSTOM** value, ensure that **CAPEXPRY** is set to the default.

CSQM532I

Queue manager *queue manager* is at version *v.r.m* and does not support attribute *attribute*.

Severity
0

Explanation

A command was issued that would result in a shared queue, or object with a **QSGDISP** value of GROUP or COPY, accessed by *queue manager*, holding a non-default value for *attribute*. IBM MQ for z/OS version *v.r.m* does not support this attribute, so it cannot be set.

System action

The command is ignored.

System programmer response

Upgrade the listed queue manager to a version of IBM MQ for z/OS that supports *attribute*.

CAPEXPRY is supported from IBM MQ for z/OS 9.4.0.

V 10.0.0 **OTELTRAC** and **OTELPCTL** are supported from IBM MQ for z/OS 9.4.3.

V 10.0.0 **OTELTRAC** and **OTELPCTL** are supported on topic objects from IBM MQ for z/OS 9.4.5.

CSQM533I

Cannot set *attribute*, as there are incompatible queue manager versions in this queue-sharing group.

Severity
0

Explanation

A command was issued that would set *attribute*, when one or more queue managers in the queue sharing group do not support it.

System action

The command is ignored.

System programmer response

Upgrade all queue managers listed in the CSQM532I messages to a version of IBM MQ for z/OS that supports *attribute*.

CAPEXPRY is supported from IBM MQ for z/OS 9.4.0.

V 10.0.0 **OTELTRAC** and **OTELPCTL** are supported from IBM MQ for z/OS 9.4.3.

V 10.0.0 **OTELTRAC** and **OTELPCTL** are supported on topic objects from IBM MQ for z/OS 9.4.5.

V 10.0.0 **CSQM534I**
csect-name Value of keyword keyword-name is unsupported due to incompatible queue manager versions in queue-sharing group *qsg-name*

Explanation
The queue manager is in a queue-sharing group and an attempt has been made to set a **CONNAME** of greater than 48 characters on a channel with a **QSGDISP** value of GROUP or COPY, but one or more queue managers in the group are earlier than IBM MQ 9.4.3.

System action
The command is ignored.

System programmer response
Either use a shorter connection name or ensure that every queue manager in the queue sharing group is at IBM MQ 9.4.3 or later, and reissue the command.

CSQM550I

csect-name Switch of transmission queue for channel *channel-name* from *old-xmitq* to *new-xmitq* started

Severity
0

Explanation

A switch of transmission queue for the channel identified by *channel-name* is required due to a change to the default cluster transmission queue configuration of the queue manager, or to the cluster channel name attribute of a cluster transmission queue. This message is issued by the queue manager when the process of switching the transmission queue from *old-xmitq* to *new-xmitq* is started.

System action

Processing continues.

System programmer response

None.

CSQM551I

csect-name Switch of transmission queue for channel *channel-name* completed - *num-msgs* messages moved from *old-xmitq* to *new-xmitq*

Severity
0

Explanation

A switch of transmission queue for the channel identified by *channel-name* was required due to a change to the default cluster transmission queue configuration of the queue manager, or to the cluster channel name attribute of a cluster transmission queue. This message is issued by the queue manager when the process of switching the transmission queue from *old-xmitq* to *new-xmitq* has completed.

During the switching process the queue manager moved *num-msgs* messages from *old-xmitq* to *new-xmitq*.

System action

Processing continues.

System programmer response

None.

CSQM552E

csect-name Switch of transmission queue for channel *channel-name* from *old-xmitq* to *new-xmitq* failed

Severity
4

Explanation

A switch of transmission queue for the channel identified by *channel-name* is required due to a change to the default cluster transmission queue configuration of the queue manager, or to the cluster channel name attribute of a cluster transmission queue. This message is issued if an error occurs when attempting to start the process of switching the transmission queue from *old-xmitq* to *new-xmitq*.

System action

The process of switching the transmission queue is not started and the channel continues to use the transmission queue *old-xmitq*.

The queue manager will retry to start the switching process the next time the channel starts.

System programmer response

Investigate why the process of switching the transmission queue could not be started, as reported in the preceding messages.

CSQM553I

csect-name Moving messages for channel *channel-name* from transmission queue *old-xmitq* to *new-xmitq*

Severity
0

Explanation

A switch of transmission queue for the channel identified by *channel-name* is required due to a change to the default cluster transmission queue configuration of the queue manager, or to the cluster channel name attribute of a cluster transmission queue. This message is issued when the process of moving messages from the old transmission queue, *old-xmitq*, to the new transmission queue, *new-xmitq*, is started.

System action

Processing continues.

System programmer response

None.

CSQM554I

csect-name Moved *num-msgs* messages for channel *channel-name* from transmission queue *old-xmitq* to *new-xmitq* - *remaining-msgs* messages remaining

Severity
0

Explanation

A switch of transmission queue for the channel identified by *channel-name* is required due to a change to the default cluster transmission queue configuration of the queue manager, or to the cluster channel name attribute of a cluster transmission queue. The switch of transmission queue requires that messages be moved from the old transmission queue, *old-xmitq*, to the new transmission queue, *new-xmitq*. This message is periodically issued to report the progress of this operation.

System action

Processing continues.

System programmer response

If this message is repeatedly issued it might indicate the old transmission queue cannot be drained of messages for the channel, which means the switching process can not complete. Applications continue to put messages to the old transmission queue during the switching process to preserve ordering. If the switching process cannot complete this might indicate that messages are being put to the old transmission queue faster than they can be moved by the switching process, or uncommitted messages remain on the old transmission queue for the channel.

CSQM555E

csect-name Moving of messages for channel *channel-name* from transmission queue *old-xmitq* to *new-xmitq* failed

Severity
8

Explanation

A switch of transmission queue for the channel identified by *channel-name* is required due to a change to the default cluster transmission queue configuration of the queue manager, or to the

cluster channel name attribute of a cluster transmission queue. The switch of transmission queue requires that messages for the channel be moved from the old transmission queue, *old-xmitq*, to the new transmission queue, *new-xmitq*. This message is issued if an error occurs while moving these messages.

System action

Moving of messages from the old transmission queue to the new transmission queue is stopped. Any existing messages on the old transmission queue and any new messages put by applications remain on the old transmission queue and are not available to be sent by the cluster-sender channel until action is taken to restart the switching process.

System programmer response

You can use preceding messages to identify and resolve the cause of the error, then restart the switching process by either stopping and starting the channel, or by using the CSQUTIL utility to restart the switching operation.

CSQM556E

csect-name Unable to open transmission queue *xmitq-name* for channel *channel-name*, MQRC=*mqrc (mqrc-text)*

Severity

8

Explanation

The switch of transmission queue requires that messages for the channel be moved from the old transmission queue to the new transmission queue. This message is issued if the old transmission queue, *xmitq-name*, cannot be opened due to reason *mqrc* when attempting to perform this operation (*mqrc-text* provides the MQRC in textual form).

System action

The switching operation fails because the moving of messages from the old transmission queue to the new transmission queue cannot be completed.

System programmer response

You can use the reason code to identify and resolve the cause of the error, then restart the switching process by either stopping and starting the channel, or by using the CSQUTIL utility to restart the switching operation. If the error cannot be resolved, or the old transmission queue has been deleted, the CSQUTIL utility can be used to perform the switching operation without moving messages from the old transmission queue to the new transmission queue. If this option is used it is the responsibility of the IBM MQ administrator to deal with any messages for this channel on the old transmission queue.

CSQM557E

csect-name Unable to open new transmission queue *xmitq-name* for channel *channel-name*, MQRC=*mqrc (mqrc-text)*

Severity

8

Explanation

The switch of transmission queue requires that messages for the channel be moved from the old transmission queue to the new transmission queue. This message is issued if the new transmission queue, *xmitq-name*, cannot be opened due to reason *mqrc* when attempting to perform this operation (*mqrc-text* provides the MQRC in textual form).

System action

The switching operation fails because the moving of messages from the old transmission queue to the new transmission queue cannot be completed.

System programmer response

You can use the reason code to identify and resolve the cause of the error, then restart the switching process by either stopping and starting the channel, or by using the CSQUTIL utility to restart the switching operation.

CSQM558E

csect-name Unable to persist transmission queue state for channel *channel-name*, MQRC=*mqrc (mqrc-text)*

Severity
8

Explanation

The queue manager uses persistent messages on the queue SYSTEM.CHANNEL.SYNCQ to track which transmission queue is used by each cluster-sender channel. This message is issued if state information cannot be updated on this queue due to reason *mqrc* (*mqrc-text* provides the MQRC in textual form).

System action

The operation requiring the persisted transmission queue state to be updated fails.

System programmer response

You can use the reason code to identify and resolve the cause of the error, then review subsequent messages to identify any additional actions that are required.

CSQM559I

csect-name Loading of cluster transmission queue state started

Severity
0

Explanation

The queue manager uses persistent messages on the queue SYSTEM.CHANNEL.SYNCQ to track which transmission queue is used by each cluster-sender channel. This message is issued during queue manager startup to indicate that loading of this information has started.

System action

Processing continues.

System programmer response

None.

CSQM560I

csect-name Loading of cluster transmission queue state completed - *num-records* records processed

Severity
0

Explanation

The queue manager uses persistent messages on the queue SYSTEM.CHANNEL.SYNCQ to track which transmission queue is used by each cluster-sender channel. This message is issued during queue manager startup to indicate loading of this information has completed. The number of cluster-sender channel records that were processed is identified by *num-records*.

System action

Processing continues.

System programmer response

None.

CSQM561E

csect-name Loading of cluster transmission queue state failed

Severity
8

Explanation

The queue manager uses persistent messages on the queue SYSTEM.CHANNEL.SYNCQ to track which transmission queue is used by each cluster-sender channel. This message is issued during queue manager startup to indicate that an error has occurred when loading this information.

System action

Processing continues with restricted clustering function. The queue manager is unable to determine which transmission queue should be used by each cluster-sender channel, so these channels are unable to start. Any requests to put a message to a remote cluster queue will fail with MQRC_CLUSTER_RESOURCE_ERROR.

System programmer response

Investigate why the transmission queue state information could not be loaded, as reported in preceding messages. Resolve the error, then restart the queue manager to restore clustering function. If you are unable to resolve the error contact IBM Support.

CSQM562E

csect-name Duplicate cluster transmission queue record found for channel *channel-name*

Severity
8

Explanation

The queue manager uses persistent messages on the queue SYSTEM.CHANNEL.SYNCQ to track which transmission queue is used by each cluster-sender channel. This message is issued during queue manager startup if a duplicate record is found for a channel.

System action

The duplicate record is ignored and processing continues, but the channel may use an incorrect transmission queue if the duplicated record should not have been used.

System programmer response

This condition should not occur. Contact IBM Support.

CSQM563E

csect-name Failed to create dynamic cluster transmission queue *xmitq-name*, MQRC=*mqrc (mqrc-text)*

Severity
4

Explanation

A channel is required to switch to a permanent-dynamic transmission queue due to a change to the default cluster transmission queue configuration of the queue manager, or to the cluster channel name attribute of a cluster transmission queue. The queue manager could not create the permanent-dynamic transmission queue, *xmitq-name*, due to reason *mqrc*.

The cluster-sender channel that is affected can be identified from the name of the transmission queue because the queue name is in the format SYSTEM.CLUSTER.TRANSMIT*channel_name*.

System action

Processing continues.

System programmer response

You can use the reason code to identify and resolve the error that has prevented the permanent-dynamic cluster transmission queue from being created. Additional messages might be issued to provide further information. If the reason code is MQRC_UNKNOWN_OBJECT_NAME this means the model queue SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE has not been defined. The definition for this model queue can be found in the supplied sample **CSQ4INSX**.

Refer to "API completion and reason codes" on page 1151 for information about *mqrc* (*mqrc-text* provides the MQRC in textual form).

CSQM564E

csect-name Cluster transmission model queue *model-xmitq* has incorrect attributes

Severity
4

Explanation

The queue manager failed to create a permanent-dynamic transmission queue for a cluster-sender channel because the model queue *model-xmitq* has been defined incorrectly.

The model queue must have the following attributes:

- The **USAGE** attribute must be set to **XMITQ**
- The **QSGDISP** attribute must not be **SHARED**
- The **DEFTYPE** attribute must be set to **PERMDYN**
- The **INDXTYPE** attribute must be set to **CORRELID**
- The **SHARE** attribute must be set

System action

Processing continues.

System programmer response

Review and correct the definition of the model transmission queue. The definition for the model queue can be found in the supplied sample **CSQ4INSX**.

CSQM565E

csect-name Delivery delay processor initialization failed, reason *reason-code*

Severity
8

Explanation

Initialization of the delivery delay processor task failed with the specified *reason-code*. As a result, any messages sent with delivery delay, using JMS 2.0, will not be processed and will be left on the delivery delay staging queue.

System action

The delivery delay processor task will end and will not restart. Messages can still be sent to the delivery delay staging queue by JMS 2.0 applications, however, the messages will not be processed until the delivery delay task is restarted. See system programmer response for instructions on how to restart the delivery delay processor.

System programmer response

The most likely reason for this message is a shortage of storage below the bar, in which case *reason-code* will be 4. Review the amount of storage used below the bar, and if possible try and reduce it. You can attempt to restart the delivery delay processor by altering the delivery delay staging queue state from 'get enabled' to 'get inhibited', and back to the 'get enabled' state again.

CSQM566I

csect-name Delivery delay processor started

Severity
0

Explanation

The delivery delay processor has started and is available to process messages from the delivery delay staging queue.

System action

Processing continues.

System programmer response

None.

CSQM567I

csect-name Delivery delay processor stopped

Severity
0

Explanation

The delivery delay processor has stopped and is no longer available to process messages from the delivery delay staging queue. This message is output in the following situations:

- The queue manager is shutting down.
- The delivery delay staging queue has been deleted, or does not exist.

System action

Processing continues.

System programmer response

None.

CSQM568E

csect-name Delivery delay processor ended abnormally, MQRC=*mqrc*

Severity
4

Explanation

The delivery delay processor has detected an error, indicated by *mqrc*, and has shut down.

System action

The delivery delay processor task ends and will not restart. Messages can still be sent to the delivery delay staging queue by JMS 2 applications, however, they will not be processed until the delivery delay task is restarted. See system programmer response for instructions on how to restart the delivery delay processor.

System programmer response

This message is output for many reasons, some of which will be expected and some will not. For example, if the delivery delay staging queue state is altered to 'get inhibited' this message will be output, and *mqrc* will be *MQRC_GET_INHIBITED*. If the message is expected then no action is required. If the message is unexpected use the value of *mqrc*, and any other messages to attempt to rectify the situation. You can attempt to restart the delivery delay processor by altering the delivery delay staging queue state from 'get enabled' to' get inhibited', and back to the 'get enabled' state again.

CSQM569I

csect-name Delivery delay processor failed to get a message with correlation ID *correlid*, MQRC=*mqrc* (*mqrc-text*)

Severity
4

Explanation

The delivery delay processor attempted to perform a destructive MQGET for the message with the specified correlation ID from the delivery delay staging queue, SYSTEM.DDELAY.LOCAL.QUEUE. The message was no longer on the queue.

System action

Processing continues.

System programmer response

Investigate whether the message was taken off the delivery delay staging queue for a valid reason, for example, it was put there by mistake. Validate the security settings for the delivery delay staging queue to ensure that only authorized users have access to it.

CSQM570E

csect-name Delivery delay processor failed to process a message with correlation ID *correlid*, for queue *q-name*, according to its report options *report-options*, MQRC=*mqrc* (*mqrc-text*)

Severity
8

Explanation

The delivery delay processor could not put the specified message to the specified queue and, so, was attempting to either send the message to the dead-letter-queue or discard the message according to the disposition options specified in the report field of the message.

The message might have also requested an exception report. Some part of the processing of the disposition options, or the report, failed with the specified return code.

System action

The message is rolled back to the delivery delay staging queue, SYSTEM.DDELAY.LOCAL.QUEUE, and reprocessed at a time decided by the system.

System programmer response

Use the information from the message to establish the cause of the problem. Some possible explanations are:

- The specified queue might no longer exist, be full, or be put disabled.
- If the message should have been put to the dead-letter-queue, check that the dead-letter-queue is defined, is not full, and is put enabled.
- If an exception report message was to be generated, check that the queue the report was to be put to is defined, is not full, is put enabled, and that the user ID in the message has access to the queue.

Otherwise, check that the dead-letter-queue is defined, is not full, and is put enabled.

CSQM571I

csect-name Delivery delay processor received an unexpected message with message ID *msgid*

Severity
4

Explanation

The delivery delay processor received a message from the delivery delay staging queue, SYSTEM.DDELAY.LOCAL.QUEUE, that was not in the correct format.

The delivery delay processor either sends the message to the dead-letter-queue, or discards it according to the disposition options specified in the report field of the message.

System action

The delivery delay processor either sends the message to the dead-letter-queue, or discards it according to the disposition options specified in the report field of the message. If the message specified an exception report, this will be generated.

System programmer response

Investigate why unexpected messages are being sent to the delivery delay staging queue. Validate the security settings for the delivery delay staging queue to ensure that only authorized users have access to it.

CSQM572E

Insufficient storage to parse data set *data set*

Severity
8

Explanation

The queue manager could not obtain sufficient storage to parse the specified data set.

System action

The queue manager terminates during start up.

System programmer response
Investigate and resolve the reason for storage shortage and restart the queue manager

CSQM573E

Unable to parse line *line* in data set *data set*

Severity
8

Explanation

The queue manager could not parse the specified line in the specified data set.

System action
The queue manager is unable to parse the specified data set but continues with initialization. It will not be possible to start TLS/SSL enabled channels.

System programmer response

Stop the queue manager and resolve any issues with the contents of the specified data set. Restart the queue manager and ensure that the queue manager is able to successfully parse the data set.

CSQM574E

Incorrect value *value* for key *key* in stanza *stanza*, data set *data set*, around line *line*

Severity
8

Explanation

The specified key (keyword) in the specified stanza and data set is not valid.

System action

The queue manager is unable to parse the specified data set but continues with initialization. It will not be possible to start TLS/SSL enabled channels.

System programmer response

Stop the queue manager and resolve any issues with the contents of the specified data set. Restart the queue manager and ensure that the queue manager is able to successfully parse the data set.

CSQM575E

Incorrect or duplicate key *key* in stanza *stanza*, data set *data set*, around line *line*

Severity
8

Explanation

The specified key (keyword) in the specified stanza and data set is either not valid or has been entered more than once.

System action

The queue manager is unable to parse the specified data set but continues with initialization. It will not be possible to start TLS/SSL enabled channels.

System programmer response

Stop the queue manager and resolve any issues with the contents of the specified data set. Restart the queue manager and ensure that the queue manager is able to successfully parse the data set.

CSQM576E

Incorrect cipher specification name *name* in key *key*, stanza *stanza* data set *data set*, around line *line*

Severity
8

Explanation

The specified cipher specification in the specified key, stanza and data set is not valid.

System action

The queue manager is unable to parse the specified data set but continues with initialization. It will not be possible to start TLS/SSL enabled channels.

System programmer response

Stop the queue manager and resolve any issues with the contents of the specified data set. Restart the queue manager and ensure that the queue manager is able to successfully parse the data set.

CSQM577E

Cipher specification name *name* in key *key*, in stanza *stanza*, data set *data set*, around line *line* is not supported

Severity
8

Explanation

The specified cipher specification in the specified key, stanza and data set is not supported.

System action

The queue manager is unable to parse the specified data set but continues with initialization. It will not be possible to start TLS/SSL enabled channels.

System programmer response

Stop the queue manager and define only supported cipher specification names in the specified data set. Restart the queue manager and ensure that the queue manager is able to successfully parse the data set.

CSQM578I

Data set *data set* read and parsed successfully

Severity
0

Explanation

The queue manager has successfully read and parsed the specified data set.

System action

The queue manager continues with initialization.

System programmer response

None.

CSQM579E

Version of z/OS does not support TLS V1.3, in key *key*, stanza *stanza*, data set *data set*, around line *line*

Severity
8

Explanation

TLS V1.3 properties have been defined in the specified key, stanza and data set however, the version of z/OS that the queue manager is being run on does not support TLS V1.3 function. To be able to use TLS V1.3 function, the z/OS operating system must be at V2.4 or higher.

System action

The queue manager is unable to parse the specified data set but continues with initialization. It will not be possible to start TLS/SSL enabled channels.

System programmer response

Stop the queue manager and remove or comment out TLS V1.3 properties from the specified data set, or upgrade to a version of the z/OS operating system that supports TLS V1.3. Restart the queue manager and ensure that the queue manager is able to successfully parse the data set.

CSQM580I

csect-name Cluster object name *object_name* located at QMID *qmid_name* is resolved using old cached information.

Severity
0

Explanation

The cluster object referenced has been resolved using old cached information.

System action

Processing continues.

System programmer response

None.

CSQM581E

Error getting data from data set *data set*

Severity
8

Explanation

The queue manager has not been able to read data from the specified data set.

System action

The queue manager is unable to parse the specified data set but continues with initialization. It will not be possible to start TLS/SSL enabled channels.

System programmer response

Stop the queue manager and resolve any issues with the contents of the data set. Restart the queue manager and ensure that the queue manager is able to successfully parse the data set.

CSQM582E

Unexpected error when accessing data set *data set*, reason *reason*

Severity
8

Explanation

The queue manager has encountered an unexpected error while accessing the specified data set.

System action

The queue manager is unable to parse the specified data set but continues with initialization. It will not be possible to start TLS/SSL enabled channels.

System programmer response

Investigate and resolve the reason for the failure. Restart the queue manager and ensure that the queue manager is able to successfully parse the data set.

If necessary, collect and gather diagnostic data, and contact IBM Support.

CSQM583E

Error during open or close of data set *data set*

Severity
8

Explanation

The queue manager encountered an error while attempting to open or close the specified data set.

System action

The queue manager is unable to parse the specified data set but continues with initialization. It will not be possible to start TLS/SSL enabled channels.

System programmer response

Stop the queue manager and ensure that the data set is defined and valid. Restart the queue manager and ensure that the queue manager is able to successfully parse the data set.

CSQM584E

Unsupported stanza skipped, stanza *stanza*, in data set *data set*

Severity
8

Explanation

The queue manager encountered an unsupported stanza in the specified data set. The stanza has been ignored.

System action

The queue manager ignores the stanza and continues to parse the remainder of the specified data set.

System programmer response

At some point, you might want to stop the queue manager, remove or comment out the unsupported stanza, restart the queue manager and ensure that the queue manager is able to successfully parse the data set.

CSQM586E

csect-name Unable to open streaming queue *streamqname* for queue *queuename*, mqrc= *mqrc (mqrc-text)*

Severity
8

Explanation

An attempt to perform an MQOPEN on the specified streaming queue failed for the specified reason, which is an IBM MQ reason code *mqrc* (*mqrc-text* provides the MQRC in textual form).

System action

The open of the streaming queue specified by the queue attribute STREAMQ does not occur.

System programmer response

For more information, see "API completion and reason codes" on page 1151. Correct the problem if required.

For a given application, the queue manager issues this message on the first failed attempt only to open the streaming queue. Subsequent failures to open the streaming queue do not result in this message being issued, unless the reason code is different.

CSQM587E

csect-name Unable to put to streaming queue *streamqname* for queue *queuename*, mqrc= *mqrc (mqrc-text)*

Severity
8

Explanation

An attempt to perform either an MQPUT or an MQPUT1 to the specified streaming queue failed for the specified reason, which is an IBM MQ reason code *mqrc* (*mqrc-text* text provides the MQRC in textual form).

System action

The put to the streaming queue specified by the queue attribute STREAMQ does not occur. Depending on the rules set by the queue attribute STRMQOS the original put might or might not have failed.

System programmer response

For more information, see "API completion and reason codes" on page 1151. Correct the problem if required.

For a given application, the queue manager issues this message on the first failed attempt only to put to the streaming queue. Subsequent failures to open the streaming queue do not result in this message being issued, unless the reason code is different.

CSQM588I

csect-name Streaming queue *qname* not supported.

Severity
4

Explanation

A command was issued to define or alter a queue with streaming enabled, and that queue does not support streaming to another queue from it.

System action

The command issued fails and does not take effect.

System programmer response

Refer to Streaming Queue Restrictions. Correct the problem if required and reissue the command.

CSQM589I

csect-name Incompatible attributes for streaming queue *qname*.

Severity
4

Explanation

A command was issued to define or alter a queue with streaming enabled, and that queue has incompatible attributes to stream from.

System action

The command issued fails and does not take effect.

System programmer response

Refer to Streaming Queue Restrictions and check the attributes of the queue. Correct the problem if required and reissue the command.

CSQM590I

csect-name STREAMQ(*streamqname*) not allowed.

Severity
4

Explanation

A command was issued to define a queue where streaming to the queue name provided in STREAMQ it is not allowed, based on its name.

System action

The command issued fails and does not take effect.

System programmer response

Refer to Streaming Queue Restrictions and check the name of the queue being used in the STREAMQ attribute. Correct the problem if required and reissue the command.

CSQM591I

csect-name Queue attributes are incompatible for use as a streaming queue.

Severity
4

Explanation

A command was issued to define or alter a queue where the queue name set in STREAMQ refers to a queue that has incompatible attributes to function as a streaming queue.

System action

The command issued fails and does not take effect.

System programmer response

Refer to Streaming Queue Restrictions and check the attributes of the queue referred to in the STREAMQ attribute. Correct the problem if required and reissue the command.

CSQM592E
csect-name OTel data not emitted to SMF. RC = *code*

Explanation
the queue manager received an non-zero return code when writing data to SMF. *code* is the hexadecimal return code from SMF. See SMFEWTM — Writing SMF records for the specific value.

System action
Processing continues, OTel data is lost. The queue manager will not emit this message again, unless it subsequently successfully writes data to SMF, and then another failure to write data occurs.

System programmer response
Ensure that SMF is enabled for OTel data, see Configuring z/OS for OTel trace. Use the value of the return code to help diagnose the problem.

CSQM999E

csect-name UNRECOGNIZED RETURN CODE *ret-code* FOR '*keyword*'

Severity
8

Explanation

An unexpected return code was issued from a command, relating to the named keyword.

System action

The command is ignored.

System programmer response

Note the return code *ret-code* (which is shown in hexadecimal) and contact IBM Support.

z/OS *Command server messages (CSQN...)*

CSQN001I

COMMAND SERVER STARTED

Severity
0

Explanation

A request to start the command server with the START CMDSERV command has been accepted.

System action

The command server is triggered to start.

CSQN002I

COMMAND SERVER ALREADY STARTED

Severity
0

Explanation

A START CMDSERV command has been entered, but the command server is already running.

System action

The command is ignored.

CSQN003I

COMMAND SERVER ENABLED

Severity
0

Explanation

In response to a START CMDSERV command in an initialization file, the command server has been put in to an enabled state.

System action

The command server will be started automatically when initialization finishes.

CSQN004I

COMMAND SERVER ALREADY ENABLED

Severity
0

Explanation

A START CMDSERV command has been entered, but the command server was already enabled.

System action

The command is ignored.

CSQN005I

COMMAND SERVER STOPPED

Severity
0

Explanation

A request to stop the command server with a STOP CMDSERV command has been accepted.

System action

The command server shuts down when it finishes processing the current command (or immediately if it is not processing a command). This message is followed by message CSQN201I to confirm that the stop has started.

CSQN006I

COMMAND SERVER ALREADY STOPPED

Severity
0

Explanation

A STOP CMDSERV command was entered, but the command server was not running.

System action

The command is ignored.

CSQN007I

COMMAND SERVER DISABLED

Severity
0

Explanation

In response to a STOP CMDSERV command in an initialization file, the command server has been put in to a disabled state.

System action

The command server will not start automatically when initialization finishes.

CSQN008I

COMMAND SERVER ALREADY DISABLED

Severity
0

Explanation

A STOP CMDSERV command has been entered, but the command server was already disabled.

System action

The command is ignored.

CSQN009I

csect-nameverb-namepkw-name COMMAND DISABLED

Severity
4

Explanation

The command was not processed because it was not allowed during this stage of initialization or termination. *verb-name* might include the command prefix (CPF). This depends on how the command was entered.

System action

The command is ignored.

CSQN011I

COMMAND SERVER STATUS IS ENABLED

Severity
0

Explanation

The command server is in an enabled state; that is, the command server will be started automatically when initialization finishes.

CSQN012I

COMMAND SERVER STATUS IS DISABLED

Severity
0

Explanation

The command server is in a disabled state; that is, the command server will not be started automatically when initialization finishes.

CSQN013I

COMMAND SERVER STATUS IS RUNNING

Severity
0

Explanation

The command server is in a running state; that is, the command server is currently processing a command.

CSQN014I

COMMAND SERVER STATUS IS WAITING

Severity
0

Explanation

The command server is in a waiting state; that is, the command server is waiting for a message to be put onto the system-command input queue.

CSQN015I

COMMAND SERVER STATUS IS STOPPED

Severity
0

Explanation

The command server is in a stopped state; that is, the command server will not process any commands until a START CMDSERV command is entered.

CSQN016I

COMMAND SERVER STATUS IS STARTING

Severity
0

Explanation

The command server is in a starting state; that is, a START CMDSERV command has been entered, but the command server has not yet started up.

CSQN017I

COMMAND SERVER STATUS IS STOPPING

Severity
0

Explanation

The command server is in a stopping state; that is, a STOP CMDSERV command has been entered, but the command server has not yet stopped.

CSQN018E

csect-name INTERNAL ERROR FOR *identifier*, RETURN CODE=*rc*

Severity
8

Explanation

This message could be caused by the following:

Identifier
Description

INSSRV01
During the early part of initialization, the queue manager was unable to start the task that processes commands in CSQINP1.

INSSRV02
During the later part of initialization, the queue manager was unable to start the task that processes commands in CSQINP2.

RTSSRV01
After initialization has completed with the command server enabled, or in response to a START CMDSERV command, the queue manager was unable to start the command server task that processes commands in the system-command input queue.

GRSSRV01
After initialization has completed with the command server enabled, or in response to a START CMDSERV command, the queue manager was unable to start the command server task that processes commands using CMDSCOPE.

System action

The task is not started.

System programmer response

Stop and restart the queue manager. Check the console for other messages regarding this error, and note the message number, *identifier*, and *rc*. Also collect the system dump (if one was produced). Contact IBM Support to report the problem.

CSQN019E

csect-name INTERNAL ERROR FOR *identifier*, RETURN CODE=*rc*

Severity

8

Explanation

This message could be caused by the following:

Identifier
Description

INSSRV01
During the early part of initialization an error occurred when trying to delete the task that processes commands in CSQINP1.

INSSRV02
During the later part of initialization an error occurred when trying to delete the task that processes commands in CSQINP2.

RTSSRV01
During termination with the command server running, or in response to a START CMDSERV command, an error occurred when trying to delete the command server task that processes commands in the system-command input queue.

GRSSRV01
During termination with the command server running, or in response to a START CMDSERV command, an error occurred when trying to delete the command server task that processes commands using CMDSCOPE.

System action

If the value of *identifier* was INSSRV01 or INSSRV02, the error is ignored, and startup continues.

If the value of *identifier* was RTSSRV01 or GRSSRV01 and *csect-name* was CSQNESTP, the command server could have terminated while processing a command.

System programmer response

Check the console for other messages regarding this error. If you are unable to resolve the problem, note the message number, *identifier*, and *rc*, collect the system dump (if one was produced), and contact IBM Support.

CSQN020E

csect-name UNABLE TO START COMMAND SERVER *identifier*

Severity

8

Explanation

csect-name was unable to start the command server task *identifier*.

System action

If *identifier* is INSSRV01 or INSSRV02, initialization is not completed and a dump might be produced. In other cases, the command server is not started.

System programmer response

Stop and restart the queue manager. Contact IBM Support with details of this message, any previous messages pertaining to this error, and the dump (if applicable).

CSQN021E

csect-name COMMAND SERVER *identifier* ABNORMAL COMPLETION

Severity
8

Explanation

The command server task *identifier* was unable to complete its processing during startup.

System action

Queue manager startup continues.

System programmer response

Check the z/OS console for related messages (probably concerning the CSQINPx data sets). The CSQOUTx data sets should also be checked to determine how much command processing was done before the error occurred. If required, reissue any unprocessed commands, or resolve the problem and restart the queue manager.

CSQN100I

COMMAND EXCEEDS MAXIMUM SIZE, COMMAND IGNORED

Severity
4

Explanation

The command string was too long.

System action

The command is ignored, and processing of CSQINP1 or CSQINP2 continues.

System programmer response

The command in question precedes this message in the CSQOUT1 or CSQOUT2 data set. For details about forming a command string, see Initialization commands.

CSQN101I

COMMAND ENDS WITH A CONTINUATION MARK, COMMAND IGNORED

Severity
4

Explanation

The last command in the CSQINP1 or CSQINP2 data set ended with a continuation mark.

System action

The command is ignored.

System programmer response

The command in question precedes this message in the CSQOUT1 or CSQOUT2 data set. For details about forming a command string, see Initialization commands.

CSQN102I

COMMAND BUFFER INVALID, ERROR UNKNOWN, COMMAND IGNORED

Severity
4

Explanation

An internal error has occurred.

System action

This command is ignored, and the next command is processed.

System programmer response

The command in question precedes this message in the CSQOUT1 or CSQOUT2 data set. If you are unable to solve the problem, contact IBM Support.

CSQN103I

COMMAND PROCESSOR RETURN CODE=*rc*, REASON CODE=*reason*

Severity

4

Explanation

An error occurred while processing the command preceding this message in the CSQOUT1 or CSQOUT2 data set. The possible values of *rc* are as follows:

Return code
Description

00000004
Internal error

00000008
Syntax or command preprocessor error, see the following lines in the CSQOUTx data set

0000000C
Command processor error, see the following lines in the CSQOUTx data set

00000010
Command processor abnormal termination

00000014
Command completed, but there is insufficient storage for the messages

00000018
Command preprocessor has insufficient storage (there could be further messages about this error)

0000001C
The command processor has insufficient storage (the command could be partially completed)

00000020
Security check

00D50102
See "Command server codes (X'D5')" on page 1034

Note: If the return code is '00000010', the reason code has no meaning.

If *reason* is 00000004 and *return code* is 00000000, the command has been accepted and will be completed later. Further messages will be produced when the command has been completed.

Otherwise the reason code indicates the command result as follows:

Reason
Description

00000000
Command completed

00000004
Partial completion

00000008
Command not actioned

0000000C
Command processor abend

FFFFFFFF
Command not actioned

System action

The next command is processed, if possible.

System programmer response

If *reason* indicates that the command did not complete, examine the command and all associated messages. See “IBM MQ for z/OS messages, completion, and reason codes” on page 289 for further information about the commands.

If you are unable to solve the problem, collect the input and output data sets and contact IBM Support.

CSQN104I

INITIALIZATION RETURN CODE=*rc*, REASON CODE=*reason*

Severity
8

Explanation

An error occurred while processing one of the initialization data sets.

System action

The system action depends on the reason code (*reason*). See “Command server codes (X'D5')” on page 1034 for information the code you have received.

System programmer response

The response you should make depends on the reason code (*reason*). See “Command server codes (X'D5')” on page 1034 for information about the code you have received.

CSQN105I

Commands from *ddname* for queue manager *qmgr-name* - *date time*

Severity
0

Explanation

This message forms the header for the output data sets CSQOUT1 and CSQOUT2.

CSQN121I
'*verb-namepkw-name*' command responses from *qmgr-name*

Explanation

The following messages are responses from queue manager *qmgr-name* to the indicated command - either entered or generated by another command - that specified CMDSCOPE.

CSQN122I
'*verb-namepkw-name*' command for CMDSCOPE(*qmgr-name*) normal completion

Explanation

Processing for the indicated command that specified CMDSCOPE(*qmgr-name*) - either entered or generated by another command - has completed successfully on all requested queue managers.

CSQN123E
'*verb-name pkw-name*' command for CMDSCOPE(*qmgr-name*) abnormal completion

Explanation

Processing for the indicated command that specified CMDSCOPE(*qmgr-name*) - either entered or generated by another command - has completed, but not successfully. If the command was sent to more than one queue manager, it might have completed successfully on some and not on others.

System programmer response

Examine the preceding responses from the command. Reissue the command correctly if necessary for the queue managers where it failed.

CSQN127E

Queue sharing group error, reason=*reason*

Severity
8

Explanation

While processing a command that specified CMDSCOPE, the command server experienced an error while trying to send data to the coupling facility.

System action

The command is not processed.

System programmer response

The response you should make depends on the reason code (*reason*). See "Coupling Facility codes (X'C5')" on page 946 for information about the code.

CSQN128E
Insufficient storage for CMDSCOPE(*qmgr-name*)

Explanation

While processing a command that specified CMDSCOPE, the command server was unable to obtain storage needed.

System action

The command is not processed.

System programmer response

If the problem persists, you might need to restart the queue manager after making more storage available.

CSQN129E

Error saving command reply information

Severity
8

Explanation

While processing a command that specified CMDSCOPE or a command for the channel initiator, the command server experienced an error while trying to save information about the command.

System action

The command is not processed.

System programmer response

The most likely cause is insufficient storage. If the problem persists, you may need to restart the queue manager after making more storage available.

CSQN130E
Command exceeds maximum size for CMDSCOPE(*qmgr-name*)

Explanation

A command that specified CMDSCOPE(*qmgr-name*) was too long.

System action

The command is not processed.

System programmer response

Reissue the command correctly.

CSQN131E
CMDSCOPE(qmgr-name) not allowed during restart

Explanation

A command that specified CMDSCOPE(*qmgr-name*) was issued in the initialization input data set CSQINP1. This is not allowed.

System action

The command is not processed.

System programmer response

Reissue the command later.

CSQN132E
CMDSCOPE(*qmgr-name*) not allowed with disposition *disposition*

Explanation

A command that specified CMDSCOPE(*qmgr-name*) with QSGDISP(*disposition*) or CHLDISP(*disposition*) was issued. This combination of values is not allowed.

System action

The command is not processed.

System programmer response

Reissue the command correctly.

CSQN133E
CMDSCOPE(*qmgr-name*) not allowed, command server unavailable

Explanation

A command that specified CMDSCOPE(*qmgr-name*) was entered or generated by another command, but the command server is not running and not enabled.

System action

The command is not processed.

System programmer response

Use the START CMDSERV command to start the command server, and reissue the command.

CSQN135E
Queue manager *qmgr-name* not active in queue sharing group

Explanation

A command specifying CMDSCOPE(*qmgr-name*) was entered or generated by another command, but that queue manager is not currently active in the group.

System action

The command is not processed.

System programmer response

Start the queue manager and reissue the command if required.

CSQN136E
Not in queue sharing group

Explanation

A command that requires a queue sharing group was entered, but the queue manager is not in a group.

System action

The command is not processed.

System programmer response

Reissue the command correctly.

CSQN137I
'*verb-name pkw-name*' accepted for CMDSCOPE(*qmgr-name*), sent to *n*

Explanation

A command that specified CMDSCOPE was entered. It has been passed to the requested queue manager(s) for processing; *n* is the number of queue managers.

System action

Processing continues.

CSQN138I
'*verb-name pkw-name*' generated for CMDSCOPE(*qmgr-name*), sent to *n*

Explanation

A command that specified CMDSCOPE was generated in response to the command originally entered. It has been passed to the indicated queue manager(s) for processing; *n* is the number of queue managers.

System action

Processing continues.

CSQN201I

COMMAND SERVER IS SHUTTING DOWN

Severity
0

Explanation

This message confirms that the command server is shutting down after an error.

System action

The command server shuts down and will not process any more commands.

System programmer response

Correct the errors reported in the preceding messages, and use the START CMDSERV command to restart the command server.

CSQN202I

COMMAND SERVER RETURN CODE=*rc*, REASON=*reason*

Severity
8

Explanation

An error occurred in the command server, as indicated by the preceding messages.

System action

The system action depends on the reason code (*reason*). See “Command server codes (X'D5')” on page 1034 or “Coupling Facility codes (X'C5')” on page 946 for information about the code.

System programmer response

The response you should make depends on the reason code (*reason*).

The return code *rc* is dependent on *reason*, and is of use to IBM service personnel.

CSQN203I

QUEUE *queuename*, MQCC=*mqcc* MQRC=*mqrc (mqrc-text)*

Severity
8

Explanation

An API call, as indicated in the preceding message, did not complete successfully. The completion code is *mqcc*, and the reason code is *mqrc* (*mqrc-text* provides the MQRC in textual form).

System programmer response

For more information, see “API completion and reason codes” on page 1151.

If you are unable to resolve the problem, note the numbers of any messages and codes associated with the error, and contact IBM Support.

Reason codes above 8000 are internal queue manager error codes. If such a code persists, report it to IBM Support.

CSQN205I

COUNT=*count*, RETURN=*rc*, REASON=*reason*

Severity
0

Explanation

This message reports the results from the command processor (refer to the section Writing programs to administer IBM MQ for further information). *count* is the number of messages (including this one) to be written to the reply-to queue in response to the command. Possible values of *rc* are as follows:

Return code
Description

00000000
Normal completion

00000004
Internal error

00000008
Syntax or command preprocessor error, see the following messages

0000000C
Command processor error, see the following messages

00000010
Command processor abnormal termination

00000014
Command completed, but there is insufficient storage for the messages

00000018
Command preprocessor has insufficient storage, (there could be further messages about this error)

0000001C
The command processor has insufficient storage (the command could be partially completed)

00000020
Security check, check userid authority

00000024
Command too long, see the following messages

00000028
Queue sharing group error, see the following messages

00D5xxxx
See “Command server codes (X'D5')” on page 1034

Note: If the return code is '00000010', the reason code has no meaning.

If *reason* is 00000004 and *return code* is 00000000, the set of reply messages is incomplete. Further sets of messages, each including another CSQN205I message, will be produced later. The results of the command will be shown by the codes in the CSQN205I message included with the final set of messages.

Otherwise the reason code indicates the command result as follows:

Reason
Description

00000000
Command completed

00000004
Partial completion

00000008
Command not actioned

0000000C
Command processor abend

FFFFFFFF
Command not actioned

System action

The next command is processed, if possible.

System programmer response

If *reason* indicates that the command did not complete, examine the command and all associated messages. See “IBM MQ for z/OS messages, completion, and reason codes” on page 289 for further information about the commands.

If you are unable to solve the problem, collect the input and output data sets and contact IBM Support.

CSQN206I

COMMAND SERVER ECBLIST, STOP=*ecb1*, WAIT=*ecb2*

Severity
8

Explanation

This message reports the ECB values associated with an error in the command server.

System action

The command server terminates.

System programmer response

This message is usually preceded by a CSQN202I message. Refer to the preceding messages for more information about the cause of the problem.

CSQN207I
COMMAND SERVER UNABLE TO OPEN REPLY TO QUEUE

Explanation

The command server was unable to open the reply-to queue while processing a command.

System action

Message CSQN203I is sent to the z/OS console reporting the completion and reason codes from the MQOPEN request.

If possible, the command server sends the response message to the dead-letter queue, otherwise the response is discarded.

System programmer response

For more information, see "API completion and reason codes" on page 1151. Use this information to solve the problem, and restart the command server. If this does not help you to solve the problem, collect the following items, and contact IBM Support.

- Return and reason codes from the message produced
- Any trace information collected

CSQN208E
COMMAND SERVER UNABLE TO OPEN COMMAND INPUT QUEUE

Explanation

The command server was unable to open the system-command input queue while starting.

System action

Message CSQN203I is sent to the z/OS console reporting the completion and reason codes from the MQOPEN request. The command server stops, without processing any commands.

System programmer response

For more information, see "API completion and reason codes" on page 1151. Use this information to solve the problem, and restart the command server. If this does not help you to solve the problem, collect the following items, and contact IBM Support.

- Return and reason codes from the message produced
- Any trace information collected

CSQN209E
COMMAND SERVER ERROR CLOSING COMMAND INPUT QUEUE

Explanation

While the command server was shutting down, an error occurred when closing the system-command input queue.

System action

Message CSQN203I is sent to the z/OS console reporting the completion and reason codes from the MQCLOSE request. The shutdown procedure continues.

System programmer response

For more information, see "API completion and reason codes" on page 1151. If this does not help you to solve the problem, collect the following items, and contact IBM Support:

- Return and reason codes from the message produced
- Any trace information collected

CSQN210E
COMMAND SERVER ERROR CLOSING REPLY TO QUEUE

Explanation

The command server was unable to close the reply-to queue while processing a command.

System action

Message CSQN203I is sent to the z/OS console reporting the completion and reason codes from the MQCLOSE request.

The command server continues.

System programmer response

For more information, see "API completion and reason codes" on page 1151.

CSQN211E
COMMAND SERVER ERROR GETTING FROM COMMAND INPUT QUEUE

Explanation

The command server experienced an error while trying to get a message from the system-command input queue.

System action

Message CSQN203I is sent to the z/OS console, reporting the completion and reason codes from the MQGET request.

The command server terminates.

System programmer response

For more information, see "API completion and reason codes" on page 1151. Use this information to solve the problem, and restart the command server. If this does not help you to solve the problem, collect the following items, and contact IBM Support:

- Return and reason codes from the console message
- Any trace information collected

CSQN212E
COMMAND SERVER ERROR PUTTING TO REPLY TO QUEUE

Explanation

The command server was unable to put a response message onto a reply-to queue while processing a command.

System action

Message CSQN203I is sent to the z/OS console reporting the completion and reason codes from the MQPUT request. If possible, the command server sends the response message to the dead-letter queue, otherwise the response is discarded.

The command server continues.

System programmer response

For more information, see "API completion and reason codes" on page 1151. If this does not help you to solve the problem, collect the following items, and contact IBM Support:

- Return and reason codes from the message produced
- Any trace information collected

CSQN213E
COMMAND SERVER ERROR, COMMAND INPUT QUEUE DISABLED

Explanation

While waiting for a command the system-command input queue has been disabled.

System action

Message CSQN203I is sent to the console containing the return and reason codes from the request function. The command server terminates.

System programmer response

Change the system-command input queue to be enabled, and issue the START CMDSERV command.

If the problem persists, collect the following items, and contact IBM Support:

- Return and reason codes
- Any trace data collected
- Printout of SYS1.LOGREC

CSQN219E

Unable to find command reply information

Severity

8

Explanation

While processing responses from a command that specified CMDSCOPE or a command for the channel initiator, the command server could not find the information to determine where to send the responses.

System action

The command might not be processed; any command responses are discarded. The command server continues.

System programmer response

If the problem persists, contact IBM Support with details of this message, any previous messages pertaining to this error, and the dump (if applicable).

CSQN220E

Error monitoring CMDSCOPE command data

Explanation

The command server experienced an error while monitoring command data in the coupling facility.

System action

Message CSQN202I is sent to the z/OS console, reporting the return and reason codes from the request.

The command server terminates.

System programmer response

See “Coupling Facility codes (X'C5')” on page 946 for information about the reason code. Use this information to solve the problem, and restart the command server. If this does not help you to solve the problem, collect the following items, and contact IBM Support:

- Return and reason codes from the console message
- Any trace information collected

CSQN221E

Error receiving CMDSCOPE command data

Explanation

The command server experienced an error while trying to get command data from the coupling facility.

System action

Message CSQN202I is sent to the z/OS console, reporting the return and reason codes from the request.

The command server terminates.

System programmer response

See “Coupling Facility codes (X'C5')” on page 946 for information about the reason code. Use this information to solve the problem, and restart the command server. If this does not help you to solve the problem, collect the following items, and contact IBM Support:

- Return and reason codes from the console message
- Any trace information collected

CSQN222E
Error sending CMDSCOPE command data

Explanation

The command server experienced an error while trying to send command data to the coupling facility.

System action

Message CSQN202I is sent to the z/OS console, reporting the return and reason codes from the request.

The command server terminates.

System programmer response

See “Coupling Facility codes (X'C5')” on page 946 for information about the reason code. Use this information to solve the problem, and restart the command server. If this does not help you to solve the problem, collect the following items, and contact IBM Support:

- Return and reason codes from the console message
- Any trace information collected

CSQN223E
Insufficient storage for CMDSCOPE command data

Explanation

The command server was unable to obtain storage needed for command data in the coupling facility.

System action

The command server terminates.

System programmer response

Use the START CMDSERV command to restart the command server. If the problem persists, you might need to restart the queue manager after making more storage available.

CSQN224E

GROUP COMMAND SERVER ENDED ABNORMALLY. RESTARTING

Severity
8

Explanation

The Group Command Server has ended abnormally because a severe error occurred.

System action

The Group Command Server is automatically restarted.

System programmer response

Investigate the reason for abnormal termination. If the problem persists contact IBM Support.

z/OS *Operations and control messages (CSQO...)*

CSQO001I

'*' may only be final character.

Severity
8

Explanation

A character string entered in the Name field contains an asterisk character that is not in the last position. This is not allowed.

System action

The main menu is redisplayed.

CSQO002I

Action *action* is not allowed.

Severity
8

Explanation

An incorrect action number was entered in the action code field. The number must be in the range shown on the panel.

System action

The panel is redisplayed.

CSQO003I

Use the ISPF command PFSHOW to display F-key settings

Severity
0

Explanation

On entry to Operations and Control, F-key settings are not being displayed. This tells you how to display the settings; you need to use F-keys to use the Operations and Control panels.

System action

None.

CSQO004I

Object *object-type* is not allowed.

Severity
8

Explanation

The value entered in the Object type field was invalid.

System action

The main menu is redisplayed.

CSQO005I

Multiple replies returned. Press F10 to view.

Severity
4

Explanation

Several error messages were returned by the queue manager in response to an action from Operations and Control.

System action

The main menu is redisplayed.

CSQO006I

Blank name is not allowed with action queue manager *.

Severity
8

Explanation

The Define action was selected and the Name field was left blank to define a new object using default attributes. However, an asterisk (*) was entered for the action queue manager, which is not allowed in this case.

System action

The main menu is redisplayed.

CSQO007I

'*field*' must be supplied.

Severity
8

Explanation

Nothing was entered in the named field. This value is required to continue.

System action

The current panel is displayed again.

CSQO008I

F-key is not active.

Severity
4

Explanation

A function key that is not currently available was pressed.

System action

The current panel is redisplayed.

CSQO009I

Action *action* is not allowed for object type *object-type*.

Severity
8

Explanation

The action number that you entered is not allowed for *object-type* objects.

System action

The current panel is redisplayed.

CSQO010I

Queue manager or group is not available.

Severity
8

Explanation

An attempt to connect to a queue manager was unsuccessful. If a queue manager name was specified, the queue manager is not running. If a queue sharing group name was specified, there are no queue managers running in that group.

System action

None, the panel is redisplayed.

CSQO011E

MQCONN unsuccessful. Reason code=*mqrc*.

Severity
8

Explanation

An attempt to connect to a queue manager or queue sharing group was unsuccessful for one of the following reasons:

1. Insufficient storage is available
2. A severe error has occurred

System action

None, the panel is redisplayed.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqrc*.

CSQO012I

Connect name is invalid or unknown.

Severity
8

Explanation

An attempt to connect to a queue manager or queue sharing group was unsuccessful because the name specified is not known, or not valid. If a blank name was specified, this means that there was no default queue manager or group defined for your installation.

System action

None, the panel is redisplayed.

CSQO013I

Not authorized to use queue manager.

Severity
8

Explanation

An attempt to connect to a queue manager was unsuccessful because the connection security failed, or you are not authorized to do so.

System action

None, the panel is redisplayed.

CSQO014E

MQOPEN of *q-name* unsuccessful. Reason code=*mqrc*.

Severity
8

Explanation

An attempt to open *q-name* was unsuccessful. *mqrc* is the reason code returned by MQOPEN; see"API completion and reason codes" on page 1151 for more information. *q-name* is one of the following:

- SYSTEM.COMMAND.INPUT
- SYSTEM.COMMAND.REPLY.MODEL; the requested dynamic queue name is appended in parentheses.
- The name of a transmission queue (if you are attempting to send commands to a remote system)

Likely causes of this problem are:

- One or both of the required queues is not defined on the queue manager that you have connected to.
- An attempt was made to send commands to a remote system, but no transport queue is defined.
- You are not authorized to open one of the required queues. If the message indicates that it is the SYSTEM.COMMAND.REPLY.MODEL queue that you are not authorized to open, it could be that you are not authorized to open the SYSTEM.CSQOREXX.* dynamic queue.
- There is insufficient storage available.

System action

The main menu is redisplayed.

CSQO015E

Command issued but no reply received.

Severity
8

Explanation

The reply to a command could not be retrieved from the reply-to queue using MQGET because the response wait time was exceeded.

System action

The panel is redisplayed. The command was sent to the queue manager, but it might not have been executed successfully.

CSQO016E

MQPUT to *q-name* unsuccessful. Reason code=*mqrc*.

Severity
8

Explanation

An attempt to put a command on a queue (*q-name*) using MQPUT was unsuccessful. *q-name* is the name of either the system-command input queue, or a transmission queue if you are sending commands to a remote queue manager. *mqrc* is the reason code returned from MQPUT; see "API completion and reason codes" on page 1151 for more information.

The most likely causes of this problem are:

1. Put requests are inhibited for the system-command input queue or the transmission queue.
2. The system-command input queue or transmission queue is full, because the command server is not running.

3. There is insufficient storage available.

System action

The command is not sent to the queue manager and the panel is redisplayed.

CSQO017E

MQGET from *reply-q* unsuccessful. Reason code=*mqrc*.

Severity
8

Explanation

The reply to a command could not be retrieved from the reply-to queue using MQGET. (The reply-to queue is a local queue generated from the model queue SYSTEM.COMMAND.REPLY.MODEL.) *mqrc* is the reason code returned from MQGET; see "API completion and reason codes" on page 1151 for more information.

A possible cause of this problem is that get requests are inhibited on the reply-to queue.

System action

The panel is redisplayed. The command was sent to the queue manager, but it might not have been executed successfully.

CSQO018E

Queue manager is invalid or unknown or unavailable.

Severity
8

Explanation

An attempt to send a command was unsuccessful because the target or action queue manager was not known or not valid or not running.

System action

The command is not sent the queue manager and the panel is redisplayed.

CSQO019E

Queue manager is no longer available.

Severity
8

Explanation

The queue manager that you were using is no longer running. The action that you requested might not have been actioned.

System action

The main menu is redisplayed.

CSQO020I

'*field*' truncated due to quotes. Press Enter to continue.

Severity
0

Explanation

The value in field *field* contains one or more quotation marks. In order that these are treated as quotation marks instead of indicators of the beginning or end of a string, each quotation mark is converted into two quotation marks (doubling up) in the command for the queue manager. However, this conversion has made the string too long, and it has been truncated.

System action

The value is truncated. The panel may be displayed again with *field-name* set to the truncated value.

CSQO021I

Generic name not allowed.

Severity
8

Explanation

You entered a name ending with an asterisk, but generic names are only allowed on the Main Menu panel.

System action

The panel is redisplayed.

CSQO022I

Filter value invalid.

Severity
8

Explanation

You asked to list objects with filtering, but the value entered for the attribute to be used was invalid.

System action

The main menu panel or an empty list panel is displayed.

CSQO023I

Command *command* not recognized.

Severity
4

Explanation

The command entered in the panel command area (or using a function key) is not valid.

System action

The panel is redisplayed.

CSQO025I

There are no messages to view.

Severity
0

Explanation

The MSGVIEW panel command was entered in the command area, or the messages function key was pressed, but there are no messages from the queue manager to view.

System action

The panel is redisplayed.

CSQO027I

Function *function* not allowed for object type *object-type*.

Severity
8

Explanation

The function number that you entered is not allowed for *object-type* objects.

System action

The current panel is redisplayed.

CSQO028I

One of '*field1*' or '*field2*' but not both must be supplied.

Severity
0

Explanation

Nothing was entered in the two named fields, or something was entered in both of them. Either one or the other must have a value.

System action

The current panel is redisplayed.

CSQO029I

Command exceeds maximum allowable length of 32762 bytes.

Severity
4

Explanation

While defining or altering a namelist, too many names are added causing the necessary command to exceed the maximum allowable length.

System action

The panel is redisplayed.

CSQO030I

No objects of type *objtype* match *name*.

Severity
0

Explanation

You asked to display or list the objects of type *objtype* and name *name*, but no matching objects were found.

System action

The current panel is redisplayed.

CSQO031E

ALLOCATE of data set *dsname* unsuccessful. Return code = *rc*.

Severity
8

Explanation

An ALLOCATE error occurred when processing the data set allocated during an attempt to edit the names in a namelist. *dsname* is the name of the data set, and is of the form *userid*.NAMELIST.NAMES*n* (where *userid* is the TSO userid involved, and *n* is a number). *rc* is the return code from the TSO command ALLOCATE.

The most likely cause of this problem is that another data set with the same name already exists, or that DDname CSQONL*n* is in use.

System action

The panel is redisplayed.

System programmer response

This message will be accompanied by one or more messages from TSO, giving more information about the cause of the problem. The return code is documented in the *TSO/E Command Reference* manual.

If you are unable to resolve the problem, contact IBM Support.

CSQO032E

Serious error returned. Press F10 to view.

Severity
12

Explanation

A command was sent to the queue manager, but message CSQN205I was received in reply, indicating a severe error.

System action

Message CSQN205I is saved. The current panel is redisplayed.

System programmer response

Look up message CSQN205I and take the appropriate action.

CSQO033E

Format of first reply not recognized. Press F10 to view.

Severity
8

Explanation

A command was sent to the queue manager, but the first reply message received is not CSQN205I.

System action

The messages received are saved. If it is not possible to continue, the current panel is redisplayed.

CSQO034E

Reply format not recognized. Press F10 to view.

Severity
8

Explanation

A command was sent to the queue manager. The first reply message received was CSQN205I as expected, but a subsequent message was not as expected.

System action

The message that caused the problem, and any subsequent messages are saved. If it is not possible to continue, the current panel is redisplayed.

CSQO035E

Unable to get storage (return code = *rc*).

Severity
12

Explanation

An attempt to get storage was unsuccessful.

System action

The system is unable to acquire enough storage.

System programmer response

Determine why there was insufficient storage available to satisfy the request.

CSQO036I

List is not filtered.

Severity
0

Explanation

You asked for a secondary list from a list that was filtered (for example, status from a list of queues or channels). The filter condition is not applied to the secondary list; all items that match the originally requested name, type, and disposition are included.

CSQO037I

Locally-defined channel will be used.

Severity
4

Explanation

You selected an action from the List Cluster queue manager Channels panel for an auto-defined cluster channel, but there is a locally-defined channel of the same name. In such a case, if you decide to take the action, it will be performed against the locally-defined channel instead.

System action

The action panel is displayed.

CSQO038I

Function is recursive.

Severity
4

Explanation

The function you requested would cause recursion; that is, it would take you to a panel that you have previously come from. This is not allowed.

System action

The current panel is redisplayed.

CSQO039E

EDIT of data set *dsname* failed. Return code = *rc*.

Severity
8

Explanation

An EDIT error occurred when processing the data set allocated during an attempt to edit the names in a namelist. *dsname* is the name of the data set, and is of the form *userid*.NAMELIST.NAMES*n* (where *userid* is the TSO userid involved, and *n* is a number). *rc* is the return code from the ISPF command EDIT.

System action

The panel is redisplayed.

System programmer response

This message will be accompanied by one or more messages from TSO, giving more information about the cause of the problem. The return code is documented in the *TSO/E Command Reference* manual.

If you are unable to resolve the problem, contact IBM Support.

CSQO040I

No open queues with disposition *disptype* match *name*.

Severity
0

Explanation

You asked to list the open queues with disposition (or dispositions) *disptype* and name *name*, but no matching objects were found.

System action

The empty list panel is displayed.

CSQO041I

Action requires a specific object type.

Severity
4

Explanation

A define request was issued for object type QUEUE or CHANNEL.

System action

The secondary window or main panel is redisplayed.

CSQO042I

On the first panel.

Severity
0

Explanation

A function key was pressed that requests scrolling back to the previous panel, but the first panel is already being displayed.

System action

The panel is redisplayed.

CSQO043I

On the last panel.

Severity
0

Explanation

A function key was pressed that requests scrolling forward to the next panel, but the last panel is already being displayed.

System action

The panel is redisplayed.

CSQO044I

Function not available for objects with type *objtype*.

Severity
0

Explanation

The function you requested (for example, status or cluster information) is not available for objects with type *objtype*.

System action

The panel is redisplayed.

CSQO045I

Name too long for object type *type*.

Severity
8

Explanation

You specified a name that was longer than 20 characters for a channel object or longer than 16 characters for a connection object or longer than 8 characters or longer than 12 characters for a CF structure object or longer than 8 characters for a storage class object.

System action

The panel is redisplayed.

CSQO046I

No channels with saved status for *name*.

Severity
0

Explanation

You asked to list the saved status for channel *name*, but there was none.

System action

The empty list panel is displayed.

CSQO047I

No current channels for *name*.

Severity
0

Explanation

You asked to list the current instances for channel *name*, but there were none.

System action

The empty list panel is displayed.

CSQO048I

Channel initiator is not active.

Severity
0

Explanation

The action you requested needs the channel initiator to be active on the action queue manager, but it is not.

System action

The panel is redisplayed.

CSQO049I

EXEC cannot be invoked as a TSO command.

Severity
4

Explanation

An attempt was made to issue one of the Operations and Control execs as a TSO command.

System action

The request is ignored.

System programmer response

Use CSQOREXX to invoke the Operations and Control panels.

CSQO050I

No objects of type *objtype* disposition *disptype* match *name*.

Severity
0

Explanation

You asked to display or list the objects of type *objtype*, with disposition (or dispositions) *disptype* and name *name*, but no matching objects were found.

System action

The current panel is redisplayed or the empty list panel is displayed.

CSQO051I

Like object name with disposition *disptype* not found. Name assumed to be for defining new object with default attributes.

Severity
0

Explanation

You asked to define an object of type *objtype*, using as a basis an object with disposition *disptype* and name *name*, but no such object was found.

(In earlier releases, you could specify the name of a new object to define on the Main Menu panel, and a 'like' name to use as a basis for your definition. Now, only the 'like' name can be specified for Define on the Main Menu panel; you specify the new object name on the Define panel.)

System action

The Define panel is displayed, initialized with the name you specified and the default attributes for that type of object, on the assumption that you intended to define a new object with default attributes.

CSQO052I

Queue manager names changed because connect name changed.

Severity
0

Explanation

The Connect name field was changed but the Target queue manager field was not, and the new connect name was different from the target queue manager name. It is assumed you have forgotten to change the target queue manager.

System action

The target queue manager is changed to the queue manager you are connected to; the action queue manager might also be changed. The 'Queue Manager Names' secondary window is displayed, showing the new names that will be used.

CSQO053I

Blank connect or queue manager names specified.

Severity
0

Explanation

One or more of Connect name, Target queue manager, or Action queue manager fields was blank, specifying that the default name should be used.

System action

The Queue Manager Names secondary window is displayed, showing the actual names that will be used.

CSQO054I

Function not available for objects with disposition *disptype*.

Severity
0

Explanation

The function you requested (for example, status or cluster information) is not available for objects with disposition (or dispositions) *disptype*.

System action

The panel is redisplayed.

CSQO055I

Connect name is a queue sharing group.

Severity
0

Explanation

The Connect name field specified the name of a queue sharing group, to connect to any queue manager in the group.

System action

The Queue Manager Names secondary window is displayed, showing the queue manager you are connected to.

CSQO056I

Queue sharing group is needed.

Severity
0

Explanation

The action you requested needs the queue manager to be part of a queue sharing group, but it is not.

System action

The panel is redisplayed.

CSQO057I

Function *function* is not allowed for disposition *disposition*.

Severity
8

Explanation

The function number that you entered is not allowed with the specified disposition. This is the disposition of the object you are working with if you are using the Manage action, or the disposition you chose if you are performing a channel function.

System action

The current panel is redisplayed.

CSQO058I

Action *action* is not allowed for channels with disposition *disposition*.

Severity
8

Explanation

The action number that you entered is not allowed for channel objects with the specified disposition.

System action

The current panel is redisplayed.

CSQO059I

Disposition *disposition* is not allowed for object type *object-type*.

Severity
8

Explanation

The disposition that you entered is not allowed for *object-type* objects.

System action

The current panel is redisplayed.

CSQO060I

Platform for target queue manager *qmgr-name* is not z/OS or OS/390®.

Severity
4

Explanation

The target queue manager is running on a plaform that is not z/OS or OS/390. With such a queue manager, it is likely that actions will work only partially, incorrectly, or not at all, and that the replies from the queue manager will not be recognized.

System action

The Confirm Target Queue Manager secondary window is displayed.

CSQO061I

Target queue manager *qmgr-name* command level is not supported.

Severity
4

Explanation

The target queue manager has a command level which is not one of those supported by the Operations and Control panels. With such a queue manager, it is likely that actions will work only partially, incorrectly, or not at all, and that the replies from the queue manager will not be recognized.

System action

The Confirm Target Queue Manager secondary window is displayed.

CSQO062I

Action queue manager *qmgr-name* command level is not the current level.

Severity
4

Explanation

The action queue manager has a command level which is not the current level supported by the Operations and Control panels. If an action is directed to such a queue manager most actions will work, but some fields will be ignored; a few objects and actions will be disallowed.

System action

The Confirm Action Queue Manager secondary window is displayed.

CSQO063I

Command level of some queue managers in the queue sharing group is not the current level.

Severity
4

Explanation

The action queue manager is * and one or more queue managers in the queue sharing group has a command level which is not the current level supported by the Operations and Control panels. If an action is directed to such a queue manager or to all queue managers in the queue sharing group, most actions will work, but some fields will be ignored; a few objects and actions will be disallowed.

System action

The Confirm Action Queue Manager secondary window is displayed.

CSQO064I

Object type *object-type* is not allowed with command level of action or target queue manager.

Severity
4

Explanation

The action or target queue manager has a command level which does not support *object-type* objects.

System action

The 'Confirm Action Queue Manager' secondary window is displayed.

CSQO065I

Object name *name* is invalid.

Severity
8

Explanation

The value entered in the Name field was invalid.

System action

The panel is redisplayed.

CSQO066I

No status of this type for CF structures matching *name*.

Severity
0

Explanation

You asked to list status for CF structures with name *name*, but there were none with status of that type.

System action

The empty list panel is displayed.

CSQO067I

Some channel initiators not active in queue sharing group. List may be incomplete.

Severity
4

Explanation

The action you requested requires information from the channel initiators on all the queue managers in the queue sharing group, but some of those channel initiators are not active. The information might therefore be incomplete.

System action

The list panel is displayed, but might be incomplete.

CSQO068I

No channel initiators active in queue sharing group.

Severity
4

Explanation

The action you requested requires information from the channel initiators on all the queue managers in the queue sharing group, but none of those channel initiators are active. No information can therefore be displayed.

System action

The empty list panel is displayed.

CSQO069I

Action or function or object type is not allowed because of queue manager command level.

Severity
4

Explanation

The action queue manager has a command level which is not the current level supported by the Operations and Control panels. The action, function, or object type you chose is not allowed at that command level.

System action

The panel is redisplayed.

CSQO070I

No field value supplied.

Severity
0

Explanation

You asked to list objects with filtering, but no value was entered into any of the fields on the filter panels. A value must be entered into one (and only one) field to specify the filtering you want.

System action

The panel is redisplayed.

CSQO071I

More than one field value supplied.

Severity
0

Explanation

You asked to list objects with filtering, but a value was entered into more than one of the fields on the filter panels. Only one field value may be entered to specify the filtering you want.

System action

The panel is redisplayed.

CSQO072I

No current channels for *name* match filter condition.

Severity
0

Explanation

You asked to list the current instances for channel *name* with a filter condition, but there were none that satisfied the condition.

System action

The empty list panel is displayed.

CSQO073I

No channels with saved status for *name* match filter condition.

Severity
0

Explanation

You asked to list the saved status for channel *name* with a filter condition, but there were none with saved status that satisfied the condition.

System action

The empty list panel is displayed.

CSQO074I

No objects of type *objtype* match *name* and filter condition.

Severity
0

Explanation

You asked to display or list the objects of type *objtype* and name *name*, with a filter condition, but no matching objects were found that satisfied the condition.

System action

The current panel is redisplayed.

CSQO075I

No objects of type *objtype* disposition *disptype* match *name* and filter condition.

Severity
0

Explanation

You asked to display or list the objects of type *objtype*, with disposition (or dispositions) *disptype* and name *name*, with a filter condition, but no matching objects were found that satisfied the condition.

System action

The current panel is redisplayed or the empty list panel is displayed.

CSQO076I

No connections match *name*.

Severity
0

Explanation

You asked to list connections with name *name*, but there were none.

System action

The empty list panel is displayed.

CSQO077I

No open handles for connection name match *name*.

Severity
0

Explanation

You asked to list the open handles for the connection *name*, but no such handles were found.

System action

The empty list panel is displayed.

CSQO078I

No connections match *name* and filter condition.

Severity
0

Explanation

You asked to list connections with name *name*, but there were none that satisfied the condition.

System action

The empty list panel is displayed.

CSQO079I

No open queues with disposition *disptype* match *name* and filter condition.

Severity
0

Explanation

You asked to list the open queues with disposition (or dispositions) *disptype* and name *name* with a filter condition, but no matching objects were found that satisfied the condition.

System action

The empty list panel is displayed.

CSQO085E

Error in *pgm-name*. TBCREATE *table-name* failed, return code = *rc*.

Severity
12

Explanation

An attempt by *pgm-name* to call the ISPF TBCREATE service was unsuccessful. *table-name* is the name of the table that *pgm-name* was attempting to create.

System action

An internal error has occurred. The current panel is redisplayed. An ISPF message giving more details about the error might be shown first.

System programmer response

An internal error has occurred, note the message number and the values contained in it, together with any associated ISPF message, and contact your IBM support center to report the problem.

CSQO086E

Error in *pgm-name*. TBDISPL *panel-name* failed, return code = *rc*.

Severity
12

Explanation

An attempt by *pgm-name* to call the ISPF TBDISPL service was unsuccessful. *panel-name* is the name of the panel that *pgm-name* was attempting to display.

System action

The system is unable to display the panel, and the last panel is redisplayed (if applicable). An ISPF message giving more details about the error might be shown first.

System programmer response

If *rc*=12, the system is unable to find the panel. If you receive this message when you are trying to display the 'Main Menu' panel it could be that you do not have the data set containing the panels in your library concatenation. Find the name of the data set containing the panels, then check your ISPPLIB library definitions. This will probably be in your TSO logon procedure unless you are calling CSQOREXX from a higher level exec or CLIST that has the ISPF LIBDEF calls in it.

If you are already using the panels when you get this message, either a panel is missing from your ISPPLIB library, or an internal error has occurred. If you are unable to solve the problem, contact IBM Support for assistance.

If *rc*=20, the most likely cause of the problem is that the system was unable to find the key-list which goes with the panel that it is trying to display. All the key lists are in an ISPF table (CSQOKEYS) that should be in a library in your ISPTLIB concatenation.

CSQO087E

Error in *pgm-name*. SELECT *program* failed, return code = *rc*.

Severity
12

Explanation

An attempt by *pgm-name* to call the ISPF SELECT service was unsuccessful. *program* is the name of the program that *pgm-name* was attempting to select.

System action

The current panel is redisplayed. An ISPF message giving more details about the error might be shown first.

System programmer response

The system is unable to find a load module. Check your ISPLLIB library concatenation.

CSQO088E

Error in *pgm-name*. DISPLAY *panel-name* failed, return code = *rc*.

Severity
12

Explanation

An attempt by *pgm-name* to call the ISPF DISPLAY service was unsuccessful. *panel-name* is the name of the panel that *pgm-name* was attempting to display.

System action

The system is unable to display the panel, and the last panel is redisplayed (if applicable). An ISPF message giving more details about the error might be shown first.

System programmer response

If *rc*=12, the system is unable to find the panel. If you receive this message when you are trying to display the 'Main Menu' panel it could be that you do not have the data set containing the panels in your library concatenation. Find the name of the data set containing the panels, then check your ISPPLIB library definitions. This will probably be in your TSO logon procedure unless you are calling CSQOREXX from a higher level exec or CLIST that has the ISPF LIBDEF calls in it.

If you are already using the panels when you get this message, either a panel is missing from your ISPPLIB library, or an internal error has occurred. If you are unable to solve the problem, contact IBM Support for assistance.

If *rc*=20, the most likely cause of the problem is that the system was unable to find the key-list which goes with the panel that it is trying to display. All the key lists are in an ISPF table (CSQOKEYS) that should be in a library in your ISPTLIB concatenation.

CSQO089E

Error in *pgm-name*. *service* failed, return code = *rc*.

Severity
12

Explanation

An attempt by *pgm-name* to call the ISPF service (*service*) was unsuccessful.

System action

The current panel is redisplayed. An ISPF message giving more details about the error might be shown first.

System programmer response

***service*=VDEFINE, VPUT, or TBADD**
An internal error has occurred, note the message number and the values contained in it, and contact IBM Support for assistance.

If *service* is anything else, note the message number and the values contained in it, together with any associated ISPF message, and contact IBM Support to report the problem.

CSQO090E

Internal error in *program*. Action field is not valid.

Severity
12

Explanation

An internal error has occurred.

System action

The current panel is redisplayed.

System programmer response

Collect the following items, and contact IBM Support:

- The number of the message, and the value of *program*
- The name of the panel involved
- A description of the actions that led to the problem

CSQO091E

Internal error in *program*. Object field is not valid.

Severity
12

Explanation

An internal error has occurred.

System action

The last panel is redisplayed.

System programmer response

Collect the following items, and contact IBM Support:

- The number of the message, and the value of *program*
- The name of the panel involved
- A description of the actions that led to the problem

CSQO092E

Internal error in *program*. Error in reply translation.

Severity
12

Explanation

An internal error has occurred.

System action

The last panel is redisplayed.

System programmer response

Collect the following items, and contact IBM Support:

- The number of the message, and the value of *program*
- The name of the panel involved
- A description of the actions that led to the problem

CSQO093E

Internal error in *program*. Command request is not valid.

Severity
12

Explanation

An internal error has occurred.

System action

The last panel is redisplayed.

System programmer response

Collect the following items, and contact IBM Support:

- The number of the message, and the value of *program*
- The name of the panel involved
- A description of the actions that led to the problem

CSQO095E

Internal error in *program*. *service* failed, return code = *rc*.

Severity
12

Explanation

An internal error has occurred.

System action

The last panel is redisplayed.

System programmer response

Collect the following items, and contact IBM Support:

- The number of the message, and the values of *program* and *service*
- The name of the panel involved
- A description of the actions that led to the problem
- Any associated ISPF message shown

CSQO096E

Internal error in *program*. *att-name* not in keyword table.

Severity
12

Explanation

An internal error has occurred.

System action

The last panel is redisplayed.

System programmer response

Collect the following items, and contact IBM Support:

- The number of the message, and the values of *program* and *att-name*
- The name of the panel involved
- A description of the actions that led to the problem

CSQO097E

Internal error in *program*. No handle for required system queue.

Severity
12

Explanation

An internal error has occurred.

System action

The last panel is redisplayed.

System programmer response

Collect the following items, and contact IBM Support:

- The number of the message
- The name of the panel involved
- A description of the actions that led to the problem

z/OS *Buffer manager messages (CSQP...)*

CSQP002I

BUFFPOOL VALUE OUT OF RANGE

Severity
8

Explanation

One of the following commands has been issued incorrectly:

- DEFINE BUFFPOOL(n)
- ALTER BUFFPOOL(n)
- DELETE BUFFPOOL(n)
- DEFINE PSID(x) BUFFPOOL(n)

The value of n is in the range 0 to 99.

System action

The command is ignored.

System programmer response

See MQSC commands for information about the command, and reissue the command correctly.

CSQP003I

PSID VALUE OUT OF RANGE

Severity
8

Explanation

One of the following commands has been issued incorrectly:

- DEFINE PSID(x)
- ALTER PSID(x)
- DELETE PSID(x)

The value of x must be in the range 0 through 99.

System action

The command is ignored.

System programmer response

See MQSC commands for information about the command, and reissue the command correctly.

CSQP004E

csect-name I/O ERROR STATUS *ret-code* PSID *psid* RBA *rba*

Severity
8

Explanation

An I/O error has occurred. *ret-code* is the return code from the Media Manager. *psid* is the identifier of the page set for which the error occurred and *rba* is the RBA (in hexadecimal) of the record on which the error occurred.

System action

The queue manager can be abended. For example, in the case of a failing MQGET or MQPUT, the queue manager is not terminated if the CSQP004E I/O error occurs during an IBM MQ API call. However, if the I/O error occurs during checkpoint processing, the queue manager is terminated.

System programmer response

See the *z/OS DFSMSdfp Diagnosis* manual for information about Media manager return codes. If you do not have access to the required manual, contact IBM Support, quoting the return code from the Media Manager.

CSQP005I

BUFFERS VALUE OUT OF RANGE

Severity
8

Explanation

One of the following commands has been issued incorrectly:

- DEFINE BUFFPOOL(n) BUFFERS(x)
- ALTER BUFFPOOL(n) BUFFERS(x)

If the value of the LOCATION parameter is BELOW, the minimum value of buffers is 100 and the maximum value is 500,000. If the value of the LOCATION parameter is ABOVE then valid values are in the range of 100 to 999999999 (nine nines).

System action

The command is ignored.

System programmer response

Reissue the command correctly. The total number of buffers that it is possible to define in all the buffer pools is determined by the amount of storage available in the queue manager address space.

CSQP006I

LOG CHECKPOINT NAME *log-name* DOES NOT MATCH QUEUE MANAGER NAME *qmgr-name*

Severity
8

Explanation

An attempt to restart with a log from another queue manager was detected. The name recorded in the log during checkpoint does not match the name of the queue manager using that log for restart.

System action

Restart is abnormally terminated with completion code X'5C6' and reason code X'00D70102'.

System programmer response

Change the started task JCL procedure xxxxMSTR for the queue manager to name the appropriate bootstrap and log data sets.

CSQP007I

Page set *x* uses buffer pool *n*

Severity
0

Explanation

This message gives the buffer pool used by the specified page set.

It is sent in response to a DEFINE PSID(x) command.

CSQP009I

PAGE RECOVERY STARTED FOR PAGE SET *psid* PAGE *page-number*

Severity
0

Explanation

An incomplete update operation was detected for page *page-number* of page set *psid*. The page is being restored to a consistent state from information on the log.

Message CSQP010I will be issued when the page recovery operation has completed.

CSQP010I

PAGE RECOVERY COMPLETE FOR PAGE SET *psid* PAGE *page-number*

Severity
0

Explanation

An incomplete update operation was detected for page *page-number* of page set *psid*. The page has been restored to a consistent state from information on the log.

CSQP011E

CONNECT ERROR STATUS *ret-code* FOR PAGE SET *psid*

Severity
8

Explanation

An attempt to open a page set was unsuccessful. *psid* is the page set identifier and *ret-code* is the return code from the Data Facilities Product (DFP) CONNECT function.

This can occur during queue manager startup, where the most likely cause is that there is no DD statement for the page set included in the queue manager started task JCL, or in response to a DEFINE PSID command used to add a page set dynamically.

System action

If this occurs during queue manager startup, MQ attempts to dynamically allocate the page set and retry the open, on the assumption that the DD statement for the page set is missing. Messages following message CSQI010I at the end of restart indicate whether the dynamic page set allocation was successful, or whether such page sets still remain offline.

If the page set cannot be opened, the queue manager continues running, but you will be unable to access the data in that page set. You could encounter problems during restart, or when attempting to open a queue.

System programmer response

If applicable, ensure that there is a DD statement for the page set included in the queue manager started task JCL.

If the page set cannot be opened, see the *z/OS DFSMSdfp Diagnosis* manual for information about Media manager return codes. If you do not have access to the required manual, contact IBM Support, quoting the return code from the Media Manager.

CSQP012I

DISCONNECT ERROR STATUS *ret-code* FOR PAGE SET *psid*

Severity
8

Explanation

An attempt to close a page set was unsuccessful. *psid* is the page set identifier and *ret-code* is the return code from the Media Manager.

System action

Queue manager shutdown continues, but some information might be missing from the page set. This will be corrected from the log during restart.

System programmer response

See the *z/OS DFSMSdfp Diagnosis* manual for information about Media manager return codes. If you do not have access to the required manual, contact your IBM support center, quoting the return code from the Media Manager.

CSQP013I

csect-name NEW EXTENT CREATED FOR PAGE SET psid. NEW EXTENT WILL NOW BE FORMATTED

Severity
0

Explanation

Page set *psid* has been successfully dynamically expanded by creating a new extent.

System action

The new extent is formatted; message CSQI031I will be issued when formatting completes successfully.

System programmer response

The page set can only be expanded 123 times. After this you will have to reallocate the page set using larger primary and secondary extents. For information about managing page sets, see Managing page sets.

CSQP014E

csect-name EXPANSION FAILED FOR PAGE SET psid. FUTURE REQUESTS TO EXTEND IT WILL BE REJECTED

Severity
8

Explanation

An attempt to expand a page set dynamically was unsuccessful.

System action

Processing continues.

System programmer response

Look for messages from VSAM or DFP that explain why the request was unsuccessful, and do the required actions.

Determine why the page set needs to expand:

- Review Planning your page sets and buffer pools to make sure your page set allocation is large enough for your application queues.
- If there is a large depth on the Dead Letter Queue (DLQ) either implement the DLQ Handler, CSQUDLQH, or clear the queue with CLEAR QLOCAL command if you don't need to take further action with the messages. Similarly, SYSTEM.EVENT.* queues can fill a page set.
- Look in joblogs or application logs to see if an error is preventing the getting application from running.
- See if an application is failing to commit its gets or puts. You can tell if there are uncommitted messages by using the following command:

```
DISPLAY QSTATUS(qname) UNCOM CURDEPTH
```

Notes:

1. The display does not show how many messages are uncommitted, and whether they are for gets or puts.
2. A message that is subject to an uncommitted MQGET still takes up space on the page set, although the message no longer contributes to the depth of the queue.

- If the getting application is a channel, is the channel starting, and is the channel able to successfully move messages? Use the command

```
DISPLAY CHSTATUS(channelname) ALL
```

to verify the channel status attributes including STATUS, SUBSTATE, and INDOUBT.

- If the messages use an integer in MQMD.EXPIRY, there might be expired messages that need to be cleaned up. If EXPRYINT is set to OFF in the QMGR definition, the command

```
REFRESH QMGR TYPE(EXPIRY) NAME(big.queue)
```

causes an EXPIRY scan of the queue that matches the name provided in the NAME() field. This command can take some time to process. Issue the command

```
DISPLAY USAGE PSID(n)
```

where n is the page set number, at regular intervals, to monitor progress.

- Check for any third party products on the system that get involved with EOV or EXTEND processing.

If you have received message IEC070I, and the *return code* (the first value in that message) is:

034(004):
End of volume - Non-extended addressable. The new allocation amount would exceed 4 GB.

If the message volume or size requires a larger page set , follow the instructions at Defining a page set to be larger than 4 GB

104
No more volumes are available on which to allocate space (no more candidate volumes).

Use the following commands to add space and switch off the internal "page not expandable" flag:

- The ALTER ADDVOLUME command; see Extending data sets on new volumes in the *z/OS DFSMS Implementing System-Managed Storage* manual for more information.
- ALTER PSID() EXPAND()

  You must supply valid syntax, that is, a page set number and expand value. See ALTER PSID for more information.

203
An extend was attempted, but no secondary space allocation quantity was specified.

204
An extend was attempted, but the maximum number of extents was reached.

The maximum number of extents for a VSAM data set cataloged in an ICF catalog is between 119 and 123, depending upon the number of extents (1-5) allocated by DADSM per allocate/extend request.

209

- An extend was attempted, but no space was available on user volume.
- No secondary space quantity was specified and no candidate volumes are available.

You can follow the directions in How to increase the size of a page set as IBM MQ for z/OS allows you to enable dynamic page set expansion, or add candidate volumes using IDCAMS ALTER ADDVOL.

The data set then needs to be closed and reopened so that the TIOT is rebuilt; otherwise IEC070I 211(8,306)-221 and IGD306I UNEXPECTED ERROR DURING IEFAB4C2 PROCESSING RETURN CODE 24 REASON CODE 0 might occur.

The close can be done without a recycle of the queue manager by using the following JCL:

```
//STEP1 EXEC PGM=IDCAMS
//DSFILE DD DSN=your.dataset.name,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
VERIFY FILE(DSFILE)
/*
```

You might need to run the JCL twice to complete with a non-zero return code. Some flags might not be reset during the first run.

Note: DFP uses up to five non-contiguous areas of disk to satisfy the total space requirements of a primary or secondary extent. This means, in the worst case of badly fragmented disk space, that you might only get around 22 times the secondary space allocated before you reach the maximum space limit.

If you believe that there is sufficient free space that could be used by another secondary extent, contact IBM Support for assistance.

220-220
IBM MQ for z/OS requested that Media Manager extend the page set (a linear data set) and passed Media Manager extend parameters that are not valid.

One cause for this is that the page set has reached the 64GB limit. Formatting page sets (FORMAT) equates a 64 GB page set with a maximum of 16,777,213 4K pages. You can observe the size of the page set using the DISPLAY USAGE command.

CSQP016E

csect-name PAGE SET *psid* HAS REACHED THE MAXIMUM NUMBER OF EXTENTS. IT CANNOT BE EXTENDED AGAIN

Severity
8

Explanation

An attempt to expand page set *psid* dynamically was unsuccessful because the maximum number of extents had been used.

System action

The page set cannot be extended again. When the messages on the full page set are retrieved, the existing space will be reused.

System programmer response

Copy the page set to a new page set with larger primary and secondary extents. By defining the page set as a multivolume data set, you can take advantage of the free space on as many disk volumes

as possible. See Planning your IBM MQ environment on z/OS. For more information about page set organization and management, see Managing page sets.

CSQP017I

csect-name EXPANSION STARTED FOR PAGE SET *psid*

Severity
0

Explanation

Page set *psid* is being expanded dynamically, by creating a new extent.

System action

All threads that are currently adding message to page set *psid* are suspended until the page set expansion completes (this is indicated by message CSQP013I).

CSQP018I

csect-name CHECKPOINT STARTED FOR ALL BUFFER POOLS

Severity
0

Explanation

A checkpoint is being taken for all defined buffer pools.

CSQP019I

csect-name CHECKPOINT COMPLETED FOR BUFFER POOL *n*, *pages* PAGES WRITTEN

Severity
0

Explanation

A checkpoint has been successfully taken for buffer pool *n*.

CSQP020E

csect-name Buffer pool *n* is too small

Severity
8

Explanation

Contention is taking place for buffers in a buffer pool. Messages will have to be read from and written to the page sets, which increases the time to process an application request and increases the amount of processor time used.

System action

Processing continues.

System programmer response

If required, use the ALTER BUFFPOOL command to add more buffers to the buffer pool. Consider first altering other buffer pools to reduce the total number of buffers in use. Refer to the latest CSQY220I message on the z/OS console to see how much virtual storage is free, and hence how many extra buffers may be safely added to a buffer pool. If you do change the number of buffers in the buffer pool, you should also change the DEFINE BUFFPOOL commands in the CSQINP1 initialization input data set used by the queue manager.

Alternatively, specify DEFINE BUFFPOOL(X) REPLACE as this option does not use the log checkpoint record.

If the buffer pool has a LOCATION value of BELOW and there is insufficient storage below the bar then consider moving the buffer above the bar by setting its LOCATION value to ABOVE. This might require altering the value of the MEMLIMIT parameter. For more information, see Address space storage.

CSQP021I

Page set *psid* new media recovery RBA=*rcvry-rba*, checkpoint RBA=*chkpt-rba*

Severity
0

Explanation

During checkpoint processing, buffers have been flushed from the buffer pools to the indicated page set, establishing a new media recovery RBA. This RBA is the point from which log data would be required to perform media recovery for the page set. It should be the same as the checkpoint RBA.

System action

Processing continues.

System programmer response

If the media recovery and checkpoint RBAs differ, contact IBM Support.

CSQP022I

Buffer pool *n* is not defined

Severity
8

Explanation

A command has been issued specifying a buffer pool that is not defined.

System action

The command is ignored.

System programmer response

See MQSC commands for information about the command, and reissue the command correctly.

CSQP023I

Request completed for buffer pool *n*, now has *k* buffers

Severity
0

Explanation

The size of the specified buffer pool has been successfully changed.

CSQP024I

Request initiated for buffer pool *n*

Severity
0

Explanation

The request to change the buffer pool has been accepted. One of the messages CSQP023I, CSQP052I, or CSQP053I will be sent to the z/OS console when the change is complete,

CSQP025I

Page set *n* is not defined or offline

Severity
8

Explanation

A command has been issued specifying a page set that is not available to the queue manager.

System action

The command is ignored.

System programmer response

See MQSC commands for information about the command, and reissue the command correctly.

CSQP026I

Page set *n* is in use by a storage class

Severity
8

Explanation

The page set specified is referenced by a storage class, and so cannot be deleted.

System action

The command is ignored.

System programmer response

Change or delete all the storage classes that reference the page set, and then reissue the command.

CSQP027I

Page set *n* has buffers in use

Severity
8

Explanation

The page set specified has buffers that are still in use, and so cannot be deleted.

System action

The command is ignored.

System programmer response

Wait until three checkpoints have been completed, and then reissue the command.

CSQP028I

Request initiated for page set *n*

Severity
0

Explanation

The request to define or delete the page set has been accepted. Message CSQP042I or CSQP032I will be sent to the z/OS console when the change is complete. If the change fails, messages CSQP041E or CSQP031E will be sent.

CSQP030E

Deallocation failed for data set *dsname*, error status=*eeeeiiii*, SMS reason code=*ssssssss*

Severity
8

Explanation

An error occurred when trying to dynamically deallocate the page set data set. Error status is the error reason code returned by z/OS dynamic allocation.

System action

The page set is deleted and is no longer available for use.

System programmer response

The error status portion of this message contains a 2-byte error code (*eeee*, S99ERROR) followed by the 2-byte information code (*iiii*, S99INFO) from the SVC99 request block. If the S99ERROR code indicates an SMS allocation error ('97xx'), then *ssssssss* contains additional SMS reason code information obtained from S99ERSN.

Go to the *z/OS MVS Authorized Assembler Services Guide* and select the Interpreting DYNALLOC return codes topic for information about these codes .

CSQP031E

Page set *n* deletion failed

Severity
8

Explanation

An error occurred while deleting the specified page set.

System action

Processing continues.

System programmer response

See the preceding error messages for more information about the error.

CSQP032I

Page set *n* deletion completed

Severity
0

Explanation

The specified page set has been successfully deleted.

CSQP033E

Error deleting page set *n*, code=*rrr*

Severity
8

Explanation

An error occurred while deleting the specified page set.

System action

The page set is not deleted, and is still available for use.

System programmer response

Note the error code and contact your IBM support center.

CSQP034E

Page set *n* is already defined

Severity
8

Explanation

The specified page set is already in use by the queue manager, and so cannot be dynamically defined.

System action

The command is ignored.

System programmer response

See MQSC commands for information about the command, and reissue the command correctly.

CSQP035E

Allocation failed for data set *dsname*, error status=*eeeeiiii*, SMS reason code=*ssssssss*

Severity
8

Explanation

An error occurred when trying to dynamically allocate the page set data set. Error status is the error reason code returned by z/OS dynamic allocation.

System action

The page set is not defined.

System programmer response

The error status portion of this message contains a 2-byte error code (*eeee*, S99ERROR) followed by the 2-byte information code (*iiii*, S99INFO) from the SVC99 request block. If the S99ERROR code indicates an SMS allocation error ('97xx'), then *ssssssss* contains additional SMS reason code information obtained from S99ERSN.

Go to the *z/OS MVS Authorized Assembler Services Guide* and select the Interpreting DYNALLOC return codes topic for information about these codes .

CSQP036I

Data set *dsname* for page set *n* is not formatted with RECOVER or REPLACE

Severity
8

Explanation

The named page set data set was not formatted correctly. A data set that is to be used for adding a page set dynamically must be one that is newly formatted (using TYPE(RECOVER)), or one that has previously been used to hold messages and has been formatted using TYPE(REPLACE).

System action

The page set is not defined.

System programmer response

Format the data set as required. If you are adding a previously unused page set to the queue manager, use the FORMAT function of the utility program CSQUTIL, specifying TYPE(RECOVER). If the page set was previously used to hold messages, use the FORMAT function specifying TYPE(REPLACE).

In the latter case, if the queue manager terminated abnormally, the formatting may fail, and message CSQU160E will be issued. It is not possible to add such a page set data set dynamically, but the page set can be brought into use again by including it in the started task JCL procedure xxxxMSTR for the queue manager, and then restarting the queue manager.

CSQP037E

OPEN failed for page set *n*, VSAM return code=*rc* reason code=*reason*

Severity
8

Explanation

A VSAM error occurred when trying to open the page set data set.

System action

The page set is not defined.

System programmer response

See the *z/OS DFSMS Macro Instructions for Data Sets* manual for information about the return and reason codes from VSAM. If necessary, reissue the request.

CSQP038E

GET failed for page set *n*, VSAM return code=*rc* reason code=*reason*

Severity
8

Explanation

A VSAM error occurred when trying to get a record from the page set data set.

System action

The page set is not defined.

System programmer response
See the *z/OS DFSMS Macro Instructions for Data Sets* manual for information about the return and reason codes from VSAM.

CSQP039E

CLOSE failed for page set *n*, VSAM return code=*rc* reason code=*reason*

Severity
8

Explanation

A VSAM error occurred when trying to close the page set data set.

System action

The page set is not defined.

System programmer response

See the *z/OS DFSMS Macro Instructions for Data Sets* manual for information about the return and reason codes from VSAM. If necessary, reissue the request.

CSQP041E

Page set *n* definition failed

Severity
8

Explanation

An error occurred while defining the specified page set.

System action

Processing continues.

System programmer response

See the preceding error messages for more information about the error.

CSQP042I

Page set *n* definition completed

Severity
0

Explanation

The specified page set has been successfully defined.

CSQP043I

Buffer pool *n* is in use by a page set

Severity
8

Explanation

The buffer pool specified is in use by a page set, and so cannot be deleted.

System action

The command is ignored.

System programmer response

Change or delete all the page sets that reference the buffer pool, and then reissue the command.

CSQP045I

Buffer pool *n* is not in use by any page set

Severity
8

Explanation

The buffer pool specified is not in use by any page set, and so cannot have buffers added or removed.

System action

The command is ignored.

System programmer response

Define at least one page set that references the buffer pool, and then reissue the command, or delete the buffer pool.

CSQP046I

Request already in progress for buffer pool *n*

Severity
8

Explanation

The buffer pool specified is being altered or deleted by another command.

System action

The command is ignored.

System programmer response

Wait until the other command has completed processing, and then reissue the command if appropriate.

CSQP047E

Unavailable page sets can cause problems - take action to correct this situation

Severity
4

Explanation

One or more page sets are unavailable, as reported in the preceding messages; they are either offline having been used previously, not defined, suspended, or otherwise inaccessible. For example, MQ

may have attempted to open a page set at restart, but failed perhaps because it was in use by another application.

This situation can cause problems, so you should take action to correct it as soon as possible.

System action

Processing continues.

System programmer response

Use the DISPLAY USAGE command to get a list of the unavailable page sets.

If a previously-used page set is required, bring it online; this can be done without stopping the queue manager. Use the FORMAT function of the utility program CSQUTIL, specifying TYPE(REPLACE). Then issue a DEFINE PSID command to bring the page set back into use. Note that all units of recovery (except those that are indoubt) that involved the offline page set will have been backed out by the queue manager when the page set was last used. These indoubt units of recovery may be resolved once the page set is back in use by the queue manager.

If a page set is not required, issue a DELETE PSID command to remove it. Also remove any DEFINE PSID command for it from the CSQINP1 initialization input data set.

CSQP048E

PUT failed for page set *n*, VSAM return code=*rc* reason code=*reason*

Severity
8

Explanation

A VSAM error occurred when trying to get a record from the page set data set.

System action

The page set is not defined.

System programmer response

See the *z/OS DFSMS Macro Instructions for Data Sets* manual for information about the return and reason codes from VSAM. If necessary, reissue the request.

CSQP049I

Data set *dsname* is formatted for a different page set *n*

Severity
8

Explanation

The page set data set was formatted using TYPE(REPLACE), and as such may contain messages for a specific page set *n*. It cannot be added dynamically with a different page set identifier.

System action

The page set is not defined.

System programmer response

Reissue the command specifying the correct data set and page set. If you intended adding a previously unused page set, reformat the data set with use the FORMAT function of the utility program CSQUTIL, specifying TYPE(RECOVER).

CSQP051I

Insufficient storage for buffer pool *n* request

Severity
4

Explanation

The size of the specified buffer pool has not been changed as requested because insufficient storage is available.

System programmer response

The DISPLAY USAGE command can be used to determine the current sizes of all buffer pools defined to the system. It may be possible to reduce the size of other buffer pools, so freeing storage, which can then be assigned to this buffer pool by reissuing the command.

Message CSQY220I shows the storage information. Refer to Managing buffer pools for more information on how to alter your buffer pool.

CSQP052I

Request partially completed for buffer pool *n*, now has *k* buffers

Severity
4

Explanation

The size of the specified buffer pool has been changed. The number of buffers is not that requested because, for example, insufficient storage is available.

CSQP053I

Request completed for buffer pool *n*, buffers not changed

Severity
0

Explanation

The size of the specified buffer pool has not been changed. This could be because the number of buffers requested was the same as the existing size, or because there was insufficient storage available to change the size or location of the buffer pool (as shown by preceding message CSQP051I).

CSQP054I

Buffer pool *n* is now located above the bar

Severity
0

Explanation

The specified buffer pool has now been moved so that it is located above the bar.

CSQP055I

Buffer pool *n* is now located below the bar

Severity
0

Explanation

The specified buffer pool has now been moved so that it is located below the bar.

CSQP056E

The ALTER BUFFPOOL command for buffer pool *n* has failed

Severity
8

Explanation

An unexpected error occurred while processing the ALTER BUFFPOOL command. The buffer pool will be left with the number of buffers that were in it at the time the error occurred.

System action

Processing continues.

System programmer response

Use the DISPLAY USAGE PSID(*) command to view the current state of the buffer pool. If necessary reissue the ALTER BUFFPOOL command again.

If any abends have been issued, look at the abend code to see if the error is caused by the queue manager being short of storage. Changing the LOCATION parameter from BELOW to ABOVE for a buffer pool might require you to increase the MEMLIMIT parameter in the JCL of the queue manager stored procedure, xxxxMSTR. For more details, see Address space storage.

If switching a buffer pool from above to below the bar you might need to decrease the number of buffers in the buffer pool.

CSQP060E

Page set 0 must use one of buffer pools 0 to 15

Severity
12

Explanation

Page set 0 must be defined so that it uses buffer pool 0 to 15.

System action

Queue manager startup fails.

System programmer response

Define page set 0 so that it uses buffer pool 0 to 15. Generally, use buffer pool 0.

CSQP061I

ALTER BUFFPOOL *n* in progress, elapsed time *m* minutes

Severity
4

Explanation

The ALTER BUFFPOOL command has been issued for the specified buffer pool. If the command takes longer than approximately five minutes to process, this message is output approximately every five minutes until the command is complete.

Once the command is complete one or more of the following messages is output: CSQP023I, CSQP051I, CSQP052I, or CSQP053I.

This message might be output in the following scenarios:

- The specified buffer pool has had its LOCATION parameter changed from ABOVE to BELOW
- The specified buffer pool had its LOCATION parameter set to ABOVE and the number of buffers has been reduced by a large number

In most cases the ALTER BUFFPOOL command completes very quickly, and this message is not output. If this message is output, it should not be a cause for concern unless the value of the elapsed time becomes a large value - more than 30 minutes.

System action

Processing continues.

System programmer response

Monitor the job log for further output of this message, or a message indicating that the ALTER BUFFPOOL command has completed.

If this message is continually output and the elapsed time grows to a large value (more than 30 minutes) this might indicate a problem, so contact IBM Support.

CSQP062I

Buffer pool *n* PAGECLAS changed, restart required to take effect

Severity
4

Explanation

The PAGECLAS attribute of the specified buffer pool has changed.

This change does not dynamically affect the type of pages used by the buffer pool, unless the LOCATION attribute is changed from BELOW to ABOVE at the same time. However the change is logged, and is applied when the queue manager is restarted.

System action

Processing continues. The buffer pool uses the previous value of the PAGECLAS attribute.

System programmer response

None, unless you require that the new PAGECLAS attribute of the specified buffer pool takes immediate effect.

In this case, either restart the queue manager or perform both of the following steps:

1. Buffer pool so that its LOCATION attribute is BELOW and its PAGECLAS is 4KB, and
2. Change the LOCATION attribute of the buffer pool to ABOVE, at the same time as changing the PAGECLAS attribute.

CSQP063E

PAGECLAS value must be 4KB if specified with LOCATION(BELOW)

Severity
8

Explanation

A buffer pool with a LOCATION value of ABOVE and PAGECLAS attribute that is not 4KB has been altered so that its LOCATION value is BELOW.

The only value of PAGECLAS that is valid with a LOCATION value of BELOW is 4KB.

System action

The command is ignored.

System programmer response

In addition to altering the LOCATION attribute to the value BELOW, alter the PAGECLAS attribute to the value 4KB.

CSQP064I

Buffer pool *n* definition in CSQINP1 data set used

Severity
4

Explanation

This message is issued at startup when the queue manager reads its log.

A buffer pool has been defined in the CSQINP1 data set, with the REPLACE attribute specified, so the definition for the buffer pool in the log of the queue manager is ignored.

Changes made to the buffer pool, using the ALTER BUFFPOOL command, when the queue manager was previously running have not occurred.

This message is only output if there is a difference between the definition for the buffer pool in the CSQINP1 data set and the log of the queue manager.

System action

The attribute values for the specified buffer pool are taken from the CSQINP1 data set rather than using the values stored in the log of the queue manager.

System programmer response

If the buffer pool definition in the CSQINP1 data set is the one you require, ignore the message.

Otherwise:

- Use the ALTER BUFFPOOL command to change the definition of the buffer pool, and also change its definition in CSQINP1 to match, or
- Remove the REPLACE attribute on the buffer pool definition in the CSQINP1 data set.

  Note, that instead of removing the REPLACE attribute you can specify the NOREPLACE attribute instead.

z/OS *IMS adapter messages (CSQQ...)*

CSQQ000I

IMS/TM *iiii* connected to queue manager *qqqq*

Severity
0

Explanation

This message is produced at the IMS master terminal when the IMS control region for IMS system *iiii* has successfully connected to queue manager *qqqq*.

CSQQ001I

IMS/TM *iiii* not connected to queue manager *qqqq*. Notify message accepted

Severity
0

Explanation

This message is produced at the IMS master terminal when the IMS control region for IMS system *iiii* has tried to connect to queue manager *qqqq* but the queue manager is not yet ready to make connections.

System action

The queue manager has accepted the notify message from IMS and when it is ready to make connections it will issue the z/OS command **MODIFY IMS** to cause IMS to attempt to make the connection again. IMS applications cannot access IBM MQ resources until the connection is made.

System programmer response

Resolve any other IBM MQ problems.

CSQQ002E

IMS/TM *iiii* failed to connect to queue manager *qqqq*, MQRC=*mqrc*

Severity
12

Explanation

This message is produced at the IMS master terminal when the IMS control region for IMS system *iiii* has failed to connect to queue manager *qqqq*. *mqrc* is the IBM MQ reason code for the failure.

System action

The IMS control region, and dependent regions are not connected to the queue manager. Any request from IMS applications for IBM MQ resources will fail.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqrc* to determine the nature of the error.

CSQQ003E

IMS/TM *iiii* create thread failed while connecting to queue manager *qqqq*, MQRC=*mqrc*

Severity

12

Explanation

This message is produced at the IMS master terminal when the IMS control region for IMS system *iiii* has failed to connect to queue manager *qqqq*. *mqrc* is the IBM MQ reason code for the failure from theIBM MQ create thread function.

System action

The IMS control region, and dependent regions are not connected to the queue manager. Any request from IMS applications for IBM MQ resources will fail.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqrc* to determine the cause of the problem.

CSQQ004E

IMS/TM *iiii* inquire indoubt failed while connecting to queue manager *qqqq*, MQRC=*mqrc*

Severity

12

Explanation

This message is produced at the IMS master terminal when the IMS control region for IMS system *iiii* has failed to connect to queue manager *qqqq*. *mqrc* is the IBM MQ reason code for the failure from the IBM MQ inquire indoubt function.

System action

The IMS control region, and dependent regions are not connected to the queue manager. Any request from IMS applications for IBM MQ resources will fail.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqrc* to determine the nature of the error.

CSQQ005E

IMS/TM *iiii* establish exit failed while connecting to queue manager *qqqq*, MQRC=*mqrc*

Severity

12

Explanation

This message is produced at the IMS master terminal when the IMS control region for IMS system *iiii* has failed to connect to queue manager *qqqq*. *mqrc* is the IBM MQ reason code for the failure from IBM MQ establish exit function.

System action

The IMS control region, and dependent regions are not connected to the queue manager. Any request from IMS applications for IBM MQ resources will fail.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqrc* to determine the cause of the error.

CSQQ007E

IMS/TM *iiii* resolve indoubt failed while connecting to queue manager *qqqq*, MQRC=*mqrc*

Severity

4

Explanation

This message is produced at the IMS master terminal when the queue manager has failed to resolve indoubt units of recovery during the connection process. *mqrc* is the IBM MQ reason code for the resolve in-doubt function failure.

System action

The IMS control region, and dependent regions are connected to the queue manager. IMS applications can access IBM MQ resources.

System programmer response

For information about resolving the IBM MQ unit of recovery associated with the in-doubt IMS unit of work, see Recovering IMS units of recovery manually.

CSQQ008I

nn units of recovery are still in doubt in queue manager *qqqq*

Severity

4

Explanation

This message is produced at the IMS master terminal when the queue manager has units of recovery still in doubt after all the IMS units of work have been resolved.

System action

The IMS control region, and dependent regions are connected to the queue manager. IMS applications can access IBM MQ resources.

System programmer response

See How in-doubt units of recovery are resolved from IMS for information about resolving the IBM MQ unit of recovery associated with the in-doubt IMS unit of work.

CSQQ010E

Error resolving unit of recovery *uuuu* (OASN *nnnn*) in queue manager *qqqq*, MQRC=*mqrc*

Severity

4

Explanation

This message is produced at the IMS master terminal when the queue manager is unable to resolve an indoubt unit of recovery. *uuuu* is the unit of work identifier in the same format as the reply from the DISPLAY THREAD command. *nnnn* is the IMS OASN (origin application sequence number), in decimal format.

System action

The IMS control region, and dependent regions are connected to the queue manager. IMS applications can access IBM MQ resources.

System programmer response

See the How in-doubt units of recovery are resolved from IMS for information about resolving the IBM MQ unit of recovery associated with the in-doubt IMS unit of work.

CSQQ011E

IMS/TM *iiii* terminate identify failed for connection to queue manager *qqqq*, MQRC=*mqrc*

Severity

12

Explanation

This message is produced at the IMS master terminal when the IMS control region for IMS system *iiii* has failed to disconnect from the queue manager *qqqq*. *mqrc* is the return code for the failure from the IBM MQ terminate identify function.

System action

The IMS control region, and dependent regions are not connected to the queue manager. Any request from IMS applications for IBM MQ resources will fail.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqrc* to determine the cause of the error.

CSQQ013I

MQ commands cannot be issued using the /SSR command

Severity

4

Explanation

This message is produced at the IMS master terminal when the /SSR IMS command is used to issue an IBM MQ command; IBM MQ commands cannot be issued in this way.

System action

None

CSQQ014E

Unit of recovery *uuuu* (OASN *nnnn*) was not committed in queue manager *qqqq*

Severity

4

Explanation

This message is produced at the IMS master terminal when, following the abnormal termination of an application, the queue manager is unable to commit an indoubt unit of recovery as requested by IMS. *uuuu* is the unit of work identifier in the same format as the reply from the DISPLAY THREAD command. *nnnn* is the IMS OASN (origin application sequence number), in decimal format.

System action

The IMS control region, and dependent regions are connected to the queue manager. IMS applications can access IBM MQ resources.

System programmer response

See How in-doubt units of recovery are resolved from IMS for information about resolving the IBM MQ unit of recovery associated with the in-doubt IMS unit of work.

CSQQ015E

Unit of recovery *uuuu* (OASN *nnnn*) was not backed out in queue manager *qqqq*

Severity
4

Explanation

This message is produced at the IMS master terminal when, following the abnormal termination of an application, the queue manager is unable to back out an indoubt unit of recovery as requested by IMS. *uuuu* is the unit of work identifier in the same format as the reply from the DISPLAY THREAD command. *nnnn* is the IMS OASN (origin application sequence number), in decimal format.

System action

The IMS control region, and dependent regions are connected to the queue manager. IMS applications can access IBM MQ resources.

System programmer response

See How in-doubt units of recovery are resolved from IMS for information about resolving the IBM MQ unit of recovery associated with the in-doubt IMS unit of work.

CSQQ100I

psb-name region-id Processing queue manager *name*

Severity
0

Explanation

This message identifies the queue manager that this instance of the IMS trigger monitor is connected to. *region-id* is the last four digits of the region identifier, or blank. This message is followed by message CSQQ110I, indicating the name of the initiation queue.

CSQQ101E

psb-name region-id Cannot open the initiation queue, MQCC=*mqcc* MQRC=*mqrc*

Severity
8

Explanation

CSQQTRMN has attempted to open an initiation queue, but the attempt was unsuccessful (for example, because the queue was not defined). *mqcc* and *mqrc* give the reason for the problem. *region-id* is the last four digits of the region identifier, or blank.

System action

CSQQTRMN ends.

System programmer response

Refer to “API completion and reason codes” on page 1151 for information about *mqcc* and *mqrc*, determine the cause of the problem, and restart CSQQTRMN.

CSQQ102E

psb-name region-id An IMS dl1-function call returned *pcb-status*

Severity
4

Explanation

A trigger message has been retrieved from the initiation queue which defines an IMS transaction to be started. However, the transaction cannot be started (for example, it cannot be found). *region-id* is the last four digits of the region identifier, or blank. *pcb-status* is the status code returned by IMS from the last *dl1-function* call.

System action

The trigger message is sent to the dead-letter queue. CSQQTRMN processes the next message.

System programmer response

See the IMS documentation for information about *pcb-status*. Examine the trigger message on the dead-letter queue to find the IMS transaction name. Determine the reason for the problem, and restart the transaction.

CSQQ103E

psb-name region-id CSQQTRMN read a trigger message with an incorrect MQTM-StrucId of *struc-id*

Severity
4

Explanation

A trigger message has been retrieved, but the structure identifier of the message is not MQTM_STRUC_ID and so is not compatible with this version of CSQQTRMN. *region-id* is the last four digits of the region identifier, or blank.

System action

The trigger message is sent to the dead-letter queue. CSQQTRMN processes the next message.

System programmer response

Check the header of the message on the dead-letter queue. This will tell you where the trigger message came from. Correct the process that created the trigger message.

CSQQ104E

psb-name region-id CSQQTRMN does not support version *version*

Severity
4

Explanation

A trigger message has been retrieved, but the version identifier in MQTM is not version 1, and so is not compatible with this version of CSQQTRMN. *region-id* is the last four digits of the region identifier, or blank.

System action

The trigger message is sent to the dead-letter queue. CSQQTRMN processes the next message.

System programmer response

Check the header of the message on the dead-letter queue. This will tell you where the trigger message came from. Correct the process that created the trigger message.

CSQQ105E

psb-name region-id CSQQTRMN cannot start a process type of *type*

Severity
4

Explanation

A trigger message has been retrieved, but the process type in MQTM is not IMS, and so cannot be processed by this version of CSQQTRMN. *region-id* is the last four digits of the region identifier, or blank.

System action

The trigger message is sent to the dead-letter queue. CSQQTRMN processes the next message.

System programmer response

Check the header of the message on the dead-letter queue. This will tell you where the trigger message came from. Correct the process that created the trigger message.

CSQQ106E

psb-name region-id MQGET error, MQCC=*mqcc* MQRC=*mqrc*. CSQQTRMN will end

Severity
8

Explanation

An attempt to issue an MQGET call on the initiation queue has been unsuccessful. *region-id* is the last four digits of the region identifier, or blank. This message is followed by message CSQQ110I, indicating the name of the queue.

System action

CSQQTRMN ends.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* to determine the cause of the problem. Restart CSQQTRMN.

CSQQ107E

psb-name region-id Cannot connect to the queue manager, MQCC=*mqcc* MQRC=*mqrc*

Severity
8

Explanation

An attempt by the trigger monitor to connect to the queue manager identified in message CSQQ100I was unsuccessful. *region-id* is the last four digits of the region identifier, or blank.

System action

CSQQTRMN ends.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* to determine the cause of the problem.

CSQQ108I

psb-name region-id LTERM *lterm-name* not available. Switched to MASTER

Severity
4

Explanation

The LTERM specified to receive diagnostic messages cannot be used.

System action

Messages are sent to the master terminal.

System programmer response

Resolve why *lterm-name* was not available.

CSQQ109E

psb-name region-id MQCLOSE error, MQCC=*mqcc* MQRC=*mqrc*

Severity
8

Explanation

An attempt has been made to close a dead-letter queue, but the MQCLOSE call was unsuccessful. *region-id* is the last four digits of the region identifier, or blank. This message is followed by message CSQQ110I, indicating the name of the queue.

System action

CSQQTRMN ends.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* to determine the cause of the problem.

CSQQ110I

Queue name = *q-name*

Severity
0

Explanation

This message follows other messages and identifies the name of the queue in question. The accompanying messages indicate the event or problem associated with the queue.

CSQQ111E

psb-name region-id CSQQTRMN read a trigger message with an incorrect length of length

Severity
4

Explanation

This message is issued if the transaction CSQQTRMN receives a trigger message that does not match the MQTM control block. *region-id* is the last four digits of the region identifier, or blank.

System action

The message is sent to the dead-letter queue.

System programmer response

Look at the message on the dead-letter queue to establish why it did not match MQTM.

CSQQ112E

psb-name region-id MQOPEN error, MQCC=*mqcc* MQRC=*mqrc*

Severity
8

Explanation

An **MQOPEN** call has been unable to open a queue. *region-id* is the last four digits of the region identifier, or blank. This message is followed by message CSQQ110I indicating the name of the queue.

System action

CSQQTRMN ends.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* to determine the cause of the problem.

CSQQ113I

psb-name region-id This message cannot be processed

Severity
0

Explanation

When an attempt to process a message using an IBM MQ API call was unsuccessful, an attempt was made to put the message on the dead-letter queue. This was also unsuccessful and the *message-id* has been sent to the LTERM. *region-id* is the last four digits of the region identifier, or blank. This message is followed by message CSQ118I, indicating the message identifier.

System action

Processing continues.

System programmer response

Check for previous messages explaining why the dead-letter queue was not available (if a dead-letter queue has not been defined, no other messages relating to the problem will have been issued).

CSQQ114E

psb-name region-id MQINQ error, MQCC=*mqcc* MQRC=*mqrc*

Severity
8

Explanation

An attempt to use the MQINQ call to inquire about the attributes of a queue was unsuccessful. *region-id* is the last four digits of the region identifier, or blank. This message is followed by message CSQQ110I indicating the name of the queue.

System action

CSQQTRMN ends.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* to determine why an MQINQ call could not be made on the queue.

CSQQ115I

psb-name region-id Ending following termination of queue manager connection

Severity
0

Explanation

CSQQTRMN has terminated because the connection to the queue manager is no longer available.

CSQQ116E

psb-name region-id Cannot open the queue manager, MQCC=*mqcc* MQRC=*mqrc*

Severity
8

Explanation

An MQOPEN call to the queue manager was unsuccessful. *region-id* is the last four digits of the region identifier, or blank.

System action

CSQQTRMN ends.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* to determine the cause of the problem.

CSQQ117E

psb-name region-id Cannot query the queue manager, MQCC=*mqcc* MQRC=*mqrc*

Severity
8

Explanation

An MQINQ call to the queue manager was unsuccessful. *region-id* is the last four digits of the region identifier, or blank.

System action

CSQQTRMN ends.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* to determine the cause of the problem.

CSQQ118I

MsgID=*msg-id*

Severity
0

Explanation

This message follows message CSQQ113I, indicating the hexadecimal identifier of the message that could not be processed.

CSQQ119E

psb-name region-id Error *rc* from STORAGE OBTAIN

Severity
8

Explanation

CSQQTRMN tried to obtain storage, but received return code *rc* from z/OS.

System action

CSQQTRMN ends.

System programmer response

Determine the reason for the return code from the STORAGE OBTAIN request, and restart CSQQTRMN.

CSQQ120E

psb-name region-id MQPUT error, MQCC=*mqcc* MQRC=*mqrc*

Severity
8

Explanation

An attempt was made to put a message on a queue with an MQPUT call, but the attempt was unsuccessful. *region-id* is the last four digits of the region identifier, or blank. This message is followed by message CSQQ110I indicating the name of the queue.

System action

CSQQTRMN ends.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* to determine why an MQPUT call could not be made for the queue.

CSQQ121E

psb-name region-id Dead-letter queue is not defined for the queue manager

Severity
4

Explanation

A dead-letter queue has not been defined for the queue manager. *region-id* is the last four digits of the region identifier, or blank.

System action

The trigger message is discarded, and the process cannot be started.

System programmer response

Define a dead-letter queue if one is required.

CSQQ122E

psb-name region-id Cannot close the queue manager, MQCC=*mqcc* MQRC=*mqrc*

Severity
8

Explanation

CSQQTRMN was unable to close the queue manager after inquiring about the dead-letter queue. *region-id* is the last four digits of the region identifier, or blank.

System action

CSQQTRMN ends.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* to determine the cause of the problem.

CSQQ123E

psb-name region-id The dead-letter queue type is not QLOCAL

Severity
4

Explanation

The dead-letter queue defined was not of type local. *region-id* is the last four digits of the region identifier, or blank. This message is followed by message CSQQ110I, indicating the name of the queue.

System action

The message is not put to the dead-letter queue.

System programmer response

Define the dead-letter queue as a local queue.

CSQQ124E

psb-name region-id The dead-letter queue usage is not NORMAL

Severity
4

Explanation

The dead-letter queue defined is not of usage type normal. *region-id* is the last four digits of the region identifier, or blank. This message is followed by message CSQQ110I, indicating the name of the queue.

System action

The message is not put to the dead-letter queue.

System programmer response

Define the dead-letter queue to have usage type normal.

CSQQ125E

psb-name region-id No initiation queue identified

Severity
8

Explanation

CSQQTRMN did not find the initiation queue name in the input parameters.

System action

CSQQTRMN ends.

System programmer response

Examine the input parameters and look for other error messages to determine the reason for the failure. Restart CSQQTRMN.

CSQQ126E

psb-name region-id An IMS *call* call returned *pcb-status*

Severity
8

Explanation

A status code of *pcb-status* was returned from a DLI call.

System action

CSQQTRMN ends.

System programmer response

Determine the reason for the status code, and restart CSQQTRMN.

CSQQ150I

csect-name IBM MQ for z/OS V*n*

Severity
0

Explanation

This message is issued as part of the header to the report issued by the IMS trigger monitor program.

CSQQ151I

csect-name Trigger Monitor Input Report - *date time*

Severity
0

Explanation

This message is issued as part of the header to the report issued by the IMS trigger monitor program.

CSQQ152I

csect-name Unable to OPEN CSQQUT1 data set

Severity
8

Explanation

The IMS trigger monitor was unable to open the data set containing input control statements.

System action

Default values are used for the options.

System programmer response

Examine the error message that has been sent to the JES log to determine the reason for the error. Check that the data set has been correctly specified.

CSQQ153I

csect-name First token is not a valid keyword

Severity
8

Explanation

The input control statement does not start with a valid keyword.

System action

The statement is ignored.

System programmer response

Correct the syntax for the statement.

CSQQ159I

csect-name Trigger monitor options:

Severity
0

Explanation

The IMS trigger monitor has finished processing input control statements. The options that will be used follow.

z/OS *Recovery manager messages (CSQR...)*

CSQR001I
RESTART INITIATED

Explanation

This message delimits the beginning of the restart process within startup. The phases of restart are about to begin. These phases are necessary to restore the operational environment to that which existed at the time of the previous termination and to perform any recovery actions that might be necessary to return IBM MQ-managed resources to a consistent state.

CSQR002I
RESTART COMPLETED

Explanation

This message delimits the completion of the restart process within startup.

System action

Startup continues.

CSQR003I
RESTART - PRIOR CHECKPOINT RBA=*rba*

Explanation

The message indicates the first phase of the restart process is in progress and identifies the log positioning RBA of the checkpoint from which the restart process will obtain its initial recovery information.

System action

Restart processing continues.

CSQR004I
RESTART - UR COUNTS - IN COMMIT=*nnnn*, INDOUBT=*nnnn*, INFLIGHT=*nnnn*, IN BACKOUT=*nnnn*

Explanation

This message indicates the completion of the first phase of the restart process. The counts indicate the number of units of recovery with an execution state during a previous queue manager termination that indicates (to ensure MQ resource consistency) some recovery action must be performed during this restart process. The counts might provide an indication of the time required to perform the remaining two phases of restart (forward and backward recovery).

The IN COMMIT count specifies the number that had started, but not completed, phase-2 of the commit process. These must undergo forward recovery to complete the commit process.

The INDOUBT count specifies the number that were interrupted between phase-1 and phase-2 of the commit process. These must undergo forward recovery to ensure that resources modified by them are unavailable until their INDOUBT status is resolved.

The INFLIGHT count specifies the number that neither completed phase-1 of the commit process nor began the process of backing out. These must undergo backward recovery to restore resources modified by them to their previous consistent state.

The IN BACKOUT count specifies the number that were in the process of backing out. These must undergo backward recovery to restore resources modified by them to their previous consistent state.

System action

Restart processing continues.

CSQR005I
RESTART - FORWARD RECOVERY COMPLETE - IN COMMIT= *nnnn*, INDOUBT=*nnnn*

Explanation

The message indicates the completion of the forward recovery restart phase. The counts indicate the number of units of recovery with recovery actions that could not be completed during the phase. Typically, those in an IN COMMIT state remain because the recovery actions of some subcomponents have not been completed. Those units of recovery in an INDOUBT state will remain until connection is made with the subsystem that acts as their commit coordinator.

System action

Restart processing continues.

CSQR006I
RESTART - BACKWARD RECOVERY COMPLETE - INFLIGHT= *nnnn*, IN BACKOUT=*nnnn*

Explanation

The message indicates the completion of the backward recovery restart phase. The counts indicate the number of units of recovery with recovery actions that could not be completed during the phase. Typically, those in either state remain because the recovery actions of some subcomponents have not been completed.

System action

Restart processing continues.

CSQR007I
UR STATUS

Explanation

This message precedes a table showing the status of units of recovery (URs) after each restart phase. The message and the table will accompany the CSQR004I, CSQR005I, or CSQR006I message after each nested phase. At the end of the first phase, it shows the status of any URs that require

processing. At the end of the second (forward recovery) and third (backout) phases, it shows the status of only those URs which needed processing but were not processed. The table helps to identify the URs that were active when the queue manager stopped, and to determine the log scope required to restart.

The format of the table is:

```
  T  CON-ID     THREAD-XREF     S   URID     TIME
```

The columns contain the following information:

T
Connection type. The values can be:

B
Batch: From an application using a batch connection

R
RRS: From an RRS-coordinated application using a batch connection

C
CICS: From CICS

I
IMS: From IMS

S
System: From an internal function of the queue manager or from the channel initiator.

CON-ID
Connection identifier for related URs. Batch connections are not related to any other connection. Subsystem connections with the same identifier indicate URs that originated from the same subsystem.

THREAD-XREF
The recovery thread cross-reference identifier associated with the thread; see Connecting from the IMS control region for more information.

S
Restart status of the UR. When the queue manager stopped, the UR was in one of these situations:

B
INBACKOUT: the UR was in the must-complete phase of backout, and is yet to be completed

C
INCOMMIT: the UR was in the must-complete phase of commit, and is yet to be completed

D
INDOUBT: the UR had completed the first phase of commit, but IBM MQ had not received the second phase instruction (the UR must be remembered so that it can be resolved when the owning subsystem reattaches)

F
INFLIGHT: the UR had not completed the first phase of commit, and will be backed out.

URID
UR identifier, the log RBA of the beginning of this unit of recovery. It is the earliest RBA required to process the UR during restart.

TIME
The time the UR was created, in the format *yyyymmdd hhmmss*. It is approximately the time of the first IBM MQ API call of the application or the first IBM MQ API call following a commit point.

CSQR009E
NO STORAGE FOR UR STATUS TABLE, SIZE REQUESTED= *xxxx*, REASON CODE=*yyyyyyyy*

Explanation

There was not enough storage available during the creation of the recoverable UR (unit of recovery) display table.

System action

Restart continues but the status table is not displayed.

System programmer response

Increase the region size of the xxxxMSTR region before restarting the queue manager.

CSQR010E
ERROR IN UR STATUS TABLE SORT/TRANSLATE, ERROR LOCATION CODE=*xxxx*

Explanation

An internal error has occurred.

System action

Restart continues but the status table is not displayed.

System programmer response

Note the error code in the message and contact IBM Support.

CSQR011E
ERROR IN UR STATUS TABLE DISPLAY, ERROR LOCATION CODE=*xxxx*

Explanation

An internal error has occurred.

System action

Restart continues but the status table is not displayed.

System programmer response

Note the error code in the message and contact IBM Support.

CSQR015E
CONDITIONAL RESTART CHECKPOINT RBA *rba* NOT FOUND

Explanation

The checkpoint RBA in the conditional restart control record, which is deduced from the end RBA or LRSN value that was specified, is not available. This is probably because the log data sets available for use at restart do not include that end RBA or LRSN.

System action

Restart ends abnormally with reason code X'00D99001' and the queue manager terminates.

System programmer response

Run the change log inventory utility (CSQJU003) specifying an ENDRBA or ENDLRSN value on the CRESTART control statement that is in the log data sets that are to be used for restarting the queue manager.

CSQR020I
OLD UOW FOUND

Explanation

During restart, a unit of work was found that predates the oldest active log. Information about the unit of work is displayed in a table in the same format as in message CSQR007I.

Old units of work can lead to extended restart times, as restart processing need to read archive logs to correctly process the unit of work. IBM MQ offers the opportunity to avoid this delay by allowing old units of work to be force committed.

Note: Force committing a unit of work can break the transactional integrity of updates between IBM MQ, and other resource managers involved in the original unit of work described in this message.

System action

Message CSQR021D is issued and the operator's reply is awaited.

CSQR021D
REPLY Y TO COMMIT OR N TO CONTINUE

Explanation

An old unit of work was found, as indicated in the preceding CSQR020I message.

System action

The queue manager waits for the operator's reply.

CSQR022I
OLD UOW COMMITTED, URID=*urid*

Explanation

This message is sent if the operator answers 'Y' to message CSQR021D.

System action

The indicated unit of work is committed.

CSQR023I
OLD UOW UNCHANGED, URID=*urid*

Explanation

This message is sent if the operator answers 'N' to message CSQR021D.

CSQR023I is also sent when an old unit of work which is already in the 'in-backout' state is identified. Units of work in the 'in-backout' state are ineligible for force commit processing as it can lead to a queue becoming unusable. For such units of work, the message CSQR021D is not issued, and no choice is possible.

System action

The indicated unit of work is left for handling by the normal restart recovery process.

CSQR026I
Long-running UOW shunted to RBA=*rba*, URID=*urid* connection name=*name*

Explanation

During checkpoint processing, an uncommitted unit of recovery was encountered that has been active for at least 3 checkpoints. The associated log records have been rewritten ('shunted') to a later point in the log, at RBA *rba*. The unit of recovery identifier *urid* together with the connection name *name* identify the associated thread.

System action

Processing continues.

System programmer response

Uncommitted units of recovery can lead to difficulties later, so consult with the application programmer to determine if there is a problem that is preventing the unit of recovery from being committed, and to ensure that the application commits work frequently enough.

CSQR027I
Long-running UOW shunting failed, URID=*urid* connection name=*name*

Explanation

During checkpoint processing, an uncommitted unit of recovery was encountered that has been active for at least 3 checkpoints. However, the associated log records could not be rewritten ('shunted') to a later point in the log. The unit of recovery identifier *urid* together with the connection name *name* identify the associated thread.

System action

The unit of recovery is not shunted, and will not participate in any future log shunting.

System programmer response

The most likely cause is insufficient active log data sets being available, in which case you should add more log data sets for the queue manager to use. Use the DISPLAY LOG command or the print log map utility (CSQJU004) to determine how many log data sets there are and what their status is.

Uncommitted units of recovery can lead to difficulties later, so consult with the application programmer to determine if there is a problem that is preventing the unit of recovery from being committed, and to ensure that the application commits work frequently enough.

CSQR029I
INVALID RESPONSE - NOT Y OR N

Explanation

The operator did not respond correctly to the reply message CSQR021D. Either 'Y' or 'N' must be entered.

System action

The original message is repeated.

CSQR030I
Forward recovery log range from RBA=*from-rba* to RBA=*to-rba*

Explanation

This indicates the log range that must be read to perform forward recovery during restart.

System action

Restart processing continues.

CSQR031I
Reading log forwards, RBA=*rba*

Explanation

This is issued periodically during restart recovery processing to show the progress of the forward recovery phase and the current status rebuild phase. For the forward recovery phase the log range that needs to be read is shown in the preceding CSQR030I message.

For the current status rebuild phase, the starting log RBA is shown in the preceding CSQR003I message and the end log RBA is shown in the preceding CSQJ099I message. The RBA represents the position in the recovery log during the forward recovery phase of current status rebuild.

System action

Restart processing continues.

CSQR032I
Backward recovery log range from RBA=*from-rba* to RBA=*to-rba*

Explanation

This indicates the log range that must be read to perform backward recovery during restart.

System action

Restart processing continues.

CSQR033I
Reading log backwards, RBA=*rba*

Explanation

This is issued periodically during restart recovery processing to show the progress of the backward recovery phase. The log range that needs to be read is shown in the preceding CSQR032I message.

System action

Restart processing continues.

CSQR034I
Backward migration detected

Explanation

During queue manager restart it has been detected that one or more of the page sets that have been connected has been used at a higher version of queue manager code.

System action

The queue manager will automatically perform special processing during restart to alter any messages stored on those page sets so they can be read by the current version of the queue manager. This special processing is dependent on there being no unresolved units of work found at the end of restart, so you might be prompted by way of further messages during restart to force commit these.

Restart processing continues.

z/OS *Topic manager messages (CSQT...)*

CSQT806I

csect-name Queued Pub/Sub Daemon started

Severity
0

Explanation

Queued Pub/Sub Daemon started

System action

None

System programmer response

None

CSQT807I

csect-name Queued Pub/Sub Daemon ended

Severity
0

Explanation

The Queued Pub/Sub Daemon has ended.

System programmer response

None

CSQT809E

csect-name Unable to process publication, Queued Pub/Sub stream queue *queue-name* is GET(DISABLED)

Severity
8

Explanation

The stream queue, *queue-name*, has been GET(DISABLED) preventing the Queued Pub/Sub Daemon from processing publication messages.

System action

The Queued Pub/Sub Daemon will continue to process publication messages on other stream queues and subscriptions on all streams.

System programmer response

To resume processing publication messages alter the stream queue to be GET(ENABLED).

To quiesce the stream remove its name from SYSTEM.QPUBSUB.QUEUE.NAMELIST.

To quiesce the Queued Pub/Sub Daemon alter the queue manager to have PSMODE(COMPAT).

CSQT810E

csect-name Unable to process subscription requests, Queued Pub/Sub control queue is GET(DISABLED)

Severity
8

Explanation

The SYSTEM.BROKER.CONTROL.QUEUE has been GET(DISABLED) preventing the Queued Pub/Sub Daemon from processing subscription requests.

System action

The Queued Pub/Sub Daemon will continue to process publication messages on stream queues.

System programmer response

To resume processing subscription requests alter the SYSTEM.BROKER.CONTROL.QUEUE to be GET(ENABLED).

To quiesce the Queued Pub/Sub Daemon alter the queue manager to have PSMODE(COMPAT).

CSQT814E

csect-name Unable to resolve parent *queue_manager_name*

Severity
8

Explanation

In establishing a publish/subscribe hierarchy, the Queued Pub/Sub Daemon has been unable to resolve the parent *queue_manager_name*.

System action

The status of the publish/subscribe parent connection will be set to error.

System programmer response

Check that the parent queue manager is correctly specified.

Ensure that broker is able to resolve the queue manager name of the parent broker.

To resolve the queue manager name, at least one of the following resources must be configured:

- A transmission queue with the same name as the parent queue manager name.
- A queue manager alias definition with the same name as the parent queue manager name.
- A cluster with the parent queue manager a member of the same cluster as this queue manager.
- A cluster queue manager alias definition with the same name as the parent queue manager name.
- A default transmission queue, modify the parent queue manager name to blank, then set with the parent queue manager name.

CSQT816E

csect-name Unable to open Queued Pub/Sub control queue MQCC=*mqcc* MQRC=*mqrc (mqrc-text)*

Severity

8

Explanation

The queue manager failed to open the Queued Publish/Subscribe control queue, SYSTEM.BROKER.CONTROL.QUEUE. The attempt to open the queue failed with completion code *mqcc* and reason *mqrc*. The most likely reasons for this error are that an application program has opened the control queue for exclusive access, or that the control queue has been defined incorrectly.

System action

The Queued Publish/Subscribe Daemon terminates.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form), then correct the problem and restart the Queued Publish/Subscribe interface.

CSQT817E

csect-name An invalid stream queue has been detected, queue *queue-name*

Severity

8

Explanation

The Pub/Sub Daemon attempted to use queue *queue-name* as a stream queue. The most likely reason for this error is that the queue is:

- Not a local queue.
- A shareable queue.
- A temporary dynamic queue.

System programmer response

Correct the problem with the queue *queue-name* or, if you do not intend to use it as a stream queue, remove it from the namelist SYSTEM.QPUBSUB.QUEUE.NAMELIST.

CSQT818E

csect-name Unable to open Queued Pub/Sub stream, queue *queue-name* MQCC=*mqcc* MQRC=*mqrc* (*mqrc-text*)

Severity

8

Explanation

The queue manager has failed to open a stream queue *queue-name*. The attempt to open the queue failed with completion code *mqcc* and reason *mqrc*. The most likely reasons for this error are:

1. A new stream name has been added to SYSTEM.QPUBSUB.QUEUE.NAMELIST but the stream queue does not exist.
2. An application has the queue open for exclusive access.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form).

CSQT819E

csect-name Queued Pub/Sub stream *stream-name* ended abnormally, reason=*mqrc*

Severity
8

Explanation

The Pub/Sub Daemon stream (*stream-name*) has ended abnormally for reason *mqrc*. The *mqrc* could be an internal return code. The queue manager will attempt to restart the stream. If the stream should repeatedly fail then the Pub/Sub Daemon will progressively increase the time between attempts to restart the stream.

System programmer response

Investigate why the problem occurred and take appropriate action to correct the problem. If the problem persists, save any generated output files and use the MQ Support site to see whether a solution is already available. If you are unable to find a match, contact IBM Support.

CSQT820E

csect-name Queued Pub/Sub stream *stream-name* restarted

Severity
8

Explanation

The queue manager has restarted a stream that ended abnormally. This message will frequently be preceded by message CSQT819E indicating why the stream ended.

System programmer response

Correct the problem.

CSQT821E

csect-name Unable to contact parent *queue_manager_name*, reason=*mqrc*

Severity
8

Explanation

In establishing a publish/subscribe hierarchy, the Queued Pub/Sub Daemon is unable to send a message to the parent *queue_manager_name* for reason *mqrc*.

System action

The status of the publish/subscribe parent connection will be set to error.

System programmer response

Investigate why the problem occurred and determine a resolution.

To reattempt a parent queue manager connection:

- Set the parent queue manager name to blank.
- Take appropriate action to correct the problem.
- Re-specify the parent queue manager name

CSQT822E

csect-name Failed to register with parent *queue_manager_name*, reason *mqrc (mqrc-text)*

Severity
8

Explanation

The Queued Pub/Sub Daemon started and the PARENT queue manager was set to *queue_manager_name* in a queue manager attribute. The queue manager attempted to register as a child of the parent, but received an exception response indicating that it was not possible. The

queue manager will retry to register periodically as a child. The child may not be able to process global publications or subscriptions correctly until this registration process has completed normally.

System programmer response

Investigate why the problem occurred and take appropriate action to correct the problem. The problem is likely to be caused by the parent queue manager not yet existing, or a problem with the transmission queue at the parent queue manager.

CSQT824I

csect-name Topic *topic-1* is dependent on PROXYSUB(FORCE) of topic *topic-2* from a different Pub/Sub hierarchy stream

Severity
4

Explanation

Topic object *topic-1* is a publish/subscribe hierarchy stream. Topic object *topic-2* is higher in the topic tree and has been configured with **PROXYSUB(FORCE)**, which results in a single wildcard proxy subscription being sent to the neighboring queue managers in the publish/subscribe hierarchy that support the *topic-2* stream. No further individual proxy subscriptions will be sent for any subscriptions made below *topic-2* in the topic tree, including below topic object *topic-1*. If a neighboring queue manager supports the *topic-1* stream, but not the *topic-2* stream, publications will not be sent to subscriptions to topic *topic-1* on this queue manager from that neighbor.

System programmer response

If the behavior described in the explanation is intended then no action is required. If not, alter the **PROXYSUB** attribute on topic *topic-1*, or *topic-2*, so both, or neither topics, are configured with the value **FORCE**.

CSQT826E

csect-name Failed to propagate subscription, stream *stream-name*, to queue manager *qm-name*, MQCC=*mqcc* MQRC=*mqrc* (*mqrc-text*)

Severity
8

Explanation

The queue manager failed to propagate subscription to stream *stream-name* at queue manager *queue_manager_name* with reason code *mqrc*. An application has either registered or unregistered a subscription to stream *stream-name*. The queue manager has attempted to propagate the subscription change to the queue manager, but the request has not been successful. Messages published on the stream through the queue manager might not reach this queue manager.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form).

Investigate why the problem occurred and take appropriate action to correct the problem.

Use the following command to refresh proxy subscriptions:

```
REFRESH QMGR TYPE(PROXYSUB)
```

CSQT827E

csect-name Queued Pub/Sub internal subscription failed. Stream *stream-name* to queue manager *queue_manager_name* reason=*reason* MQRC= *mqrc*

Severity
8

Explanation

The queue manager failed to subscribe to stream *stream-name* at queue manager *queue_manager_name* with reason code *mqrc*. Related queue managers learn about each others configuration by subscribing to information published by each other. A queue manager discovered that one of these internal subscriptions has failed. The queue manager will reissue the subscription immediately. The queue manager cannot function correctly without knowing some information about neighboring queue managers. The information that this broker has about queue manager *queue_manager_name* is not complete and this could lead to subscriptions and publications not being propagated around the network correctly.

System programmer response

Investigate why the problem occurred and take appropriate action to correct the problem. The most likely cause of this failure is a problem with the transmission queue at the queue manager *queue_manager_name* or a problem with the definition of the route between this queue manager and queue manager *queue_manager_name*

CSQT831E

csect-name Unable to make subscription, reason=*mqrc* (*mqrc-text*), subscription name *sub-name*, topic *topic-string*

Severity
8

Explanation

A failure occurred while attempting to create a subscription to topic string *topic-string* using the subscription name *sub-name*. The associated reason code is *mqrc*. The *mqrc* could be an internal return code.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form).

CSQT833E

csect_name Queue manager *queue_manager_name* introduced a loop into the Pub/Sub hierarchy

Severity
8

Explanation

The queue manager *queue_manager_name* introduced a loop in the Pub/Sub hierarchy. The Queued Publish/Subscribe Daemon on this queue manager will terminate immediately.

System programmer response

Remove queue manager *queue_manager_name* from the hierarchy, either by deleting the queue manager, or by removing knowledge of the queue manager's parent, using the ALTER QMGR PARENT(' ') command, or in exceptional circumstances, RESET QMGR TYPE(PUBSUB) PARENT(*queue_manager_name*).

CSQT834E

csect-name Conflicting queue manager names in the Pub/Sub hierarchy

Severity
8

Explanation

The names of the queue managers (*queue_manager_name*) and (*queue_manager_name*) in the Pub/Sub hierarchy both start with the same 12 characters. The first 12 characters of a queue manager name should be unique to ensure that no confusion arises within the hierarchy, and to guarantee unique message ID allocation.

CSQT835E

csect-name Unable to inform parent *parent-name* of new relation *queue_manager_name*, reason=*mqrc (mqrc-text)*

Severity
8

Explanation

The queue manager failed to notify its parent queue manager *parent-name* of the relation *queue_manager_name* in the Pub/Sub hierarchy. The notification message will be put to the parent's dead-letter queue. A failure to notify a queue manager of a new relation will mean that no loop detection can be performed for the new relation.

System programmer response

Diagnose and correct the problem on the parent queue manager. One possible reason for this is that the parent queue manager does not yet exist.

CSQT836E

csect-name Duplicate queue manager name *queue_manager_name* located in the Pub/Sub hierarchy

Severity
8

Explanation

Multiple instances of the queue manager name *queue_manager_name* have been located. This could either be the result of a previously resolved loop in the Pub/Sub hierarchy, or multiple queue managers in the Pub/Sub hierarchy having the same name.

System programmer response

If this queue manager introduced a loop in the hierarchy (typically identified by message CSQT833E), this message can be ignored. It is strongly recommended that every queue manager in a Pub/Sub hierarchy has a unique name. It is not recommended that multiple queue managers use the same name.

CSQT839E

csect-name Unexpected topology information received from queue manager *queue_manager_name*

Severity
8

Explanation

A queue manager has received a distributed publish/subscribe communication that it did not expect. The message was sent by queue manager *queue_manager_name*. The message will be processed according to the report options in that message. The most likely reason for this message is that the queue manager topology has been changed while distributed publish/subscribe communication messages were in transit (for example, on a transmission queue) and that a message relating to the previous queue manager topology has arrived at a queue manager in the new topology. This message may be accompanied by an informational FFST including details of the unexpected communication.

System programmer response

If the queue manager topology has changed and the queue manager named in the message is no longer related to the queue manager issuing this message, this message can be ignored. If the **RESET QMGR TYPE(PUBSUB)** command was issued to unilaterally remove knowledge of queue manager *queue_manager_name* from this queue manager, the **RESET QMGR TYPE(PUBSUB)** command should also be used to remove knowledge of this queue manager from queue manager *queue_manager_name*.

CSQT844E

csect-name The relation with *queue_manager_name* is unknown

Severity
8

Explanation

The RESET QMGR TYPE(PUBSUB) command has been issued in an attempt to remove a queue manager's knowledge of a relation of that queue manager. The relative *queue_manager_name* is unknown at queue manager *queue_manager_name*. If the parent KEYWORD was specified, the queue manager does not currently have a parent. If the CHILD keyword was specified, the queue manager does not recognize the named child.

System programmer response

Investigate why the queue manager is unknown.

CSQT848E

csect-name Failed to register proxy subscription for queue manager *qmgr-name*, stream *stream-name*, topic string *topic-string*, reason=*mqrc* (*mqrc-text*)

Severity
8

Explanation

The queue manager received a proxy subscription request for stream *stream-name* and topic *topic-string* from queue manager *qmgr-name*. The attempt to register the subscription was unsuccessful for reason *mqrc* (*mqrc-text* provides the MQRC in textual form). Messages published upon this topic will not be delivered to subscriptions on the relation queue manager.

System programmer response

Use the reason code to investigate why the failure occurred and take appropriate action to correct the problem. Use the command REFRESH QMGR TYPE(PROXYSUB) on the relation queue manager to refresh it's proxy subscriptions.

CSQT852E

csect-name Unable to propagate delete publication command, topic *topic-name*, stream *stream-name*, to queue manager *queue_manager_name*, reason=*mqrc* (*mqrc-text*)

Severity
8

Explanation

The queue manager failed to propagate delete publication command for stream *stream-name* to related queue manager *queue_manager_name* for reason *mqrc*. When an application issues a delete publication command to delete a global publication, the command has to be propagated to all queue managers in the sub-hierarchy supporting the stream. The queue manager reporting the error has failed to forward a delete publication command to a related queue manager *queue_manager_name* who supports stream *stream-name*. Delete publication commands are propagated without MQRO_DISCARD_MSG and the command message might have been written to a dead-letter queue. The topic for which the delete publication has failed is *topic-name*.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form).

If the delete publication has failed because the stream has been deleted at the related queue manager, this message can be ignored. Investigate why the delete publication has failed and take the appropriate action to recover the failed command.

CSQT853E

csect-name Unable to propagate delete publication command, topic *topic-name*, stream *stream-name*, relation *relation-name*, reason = *mqrc* (*mqrc-text*)

Severity
8

Explanation

The queue manager failed to propagate a delete publication command for stream *stream-name* to a previously related queue manager *relation-name*. In some cases the stream or the relation cannot be determined and so is shown as '????'.

When an application issues a delete publication command to delete a global publication, the command is propagated to all queue managers in the sub-hierarchy supporting the stream. The queue manager topology was changed after deleting the publication, but before a queue manager removed by the topology change processed the propagated delete publication message. The topic for which the delete publication has failed is *topic-name*. In some cases the topic cannot be determined and so is shown as '????'.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form).

It is the user's responsibility to quiesce queue manager activity before changing the queue manager topology using the RESET QMGR TYPE(PUBSUB) command. Investigate why this delete publication activity was not quiesced. The delete publication command will have been written to the dead-letter queue at the queue manager that was removed from the topology. In this case, further action might be necessary to propagate the delete publication command that was not quiesced before the RESET QMGR TYPE(PUBSUB) command was issued.

CSQT854E

csect-name Unable to propagate delete publication command, topic *topic-name*, stream *stream-name* to queue manager *queue_manager_name*

Severity
8

Explanation

When an application issues a delete publication command, the command has to be propagated to all queue managers in the sub-hierarchy supporting the stream. At the time the delete publication was propagated, queue manager *queue_manager_name* was a known relation of this message queue manager supporting stream *stream-name*. Before the delete publication command arrived at the related queue manager, the queue manager topology was changed so that queue manager *queue_manager_name* no longer supported stream *stream-name*. The topic for which the delete publication has failed is *topic-name*.

System programmer response

It is the user's responsibility to quiesce queue manager activity before changing the stream topology of the queue manager. Investigate why this delete publication activity was not quiesced. The delete publication command will have been written to the dead-letter queue at queue manager *queue_manager_name*.

CSQT855E

csect-name Queued Pub/Sub Daemon failed, reason=*mqrc*

Severity
8

Explanation

An attempt has been made to run the queued publish/subscribe interface (Queued Pub/Sub Daemon) but the interface has ended for reason *mqrc*.

If *mqrc* is a number in the range of 2000 - 3000, it is an API reason code. If it is of the form 5*nnn*, it is a queued publish/subscribe message code associated with the message CSQT *nnn*E, which is normally issued previously.

System programmer response

If *mqrc* is an API reason code, see "API completion and reason codes" on page 1151 for more information about the *mqrc*. If *mqrc* is a queued publish/subscribe message code, see the corresponding message explanation for more information. Where no such message exists, see "Queued Publish/Subscribe message codes" on page 1150 for the corresponding message number.

Determine why the queued publish/subscribe daemon ended. The message logs for the Channel Initiator might contain more detailed information about why the queued publish/subscribe daemon cannot be started. Resolve the problem that is preventing the daemon from completing and restart the Channel Initiator.

CSQT856E

csect-name Unable to process publish command message for stream *stream-name*, reason=*mqrc (mqrc-text)*

Severity
8

Explanation

The Queued Pub/Sub Daemon failed to process a publish message for stream *stream-name*. The queue manager was unable to write the publication to the dead-letter queue and was not permitted to discard the publication. The queue manager will temporarily stop the stream and will restart the stream and consequently retry the publication after a short interval.

System programmer response

Investigate why the error has occurred and why the publication cannot be written to the dead-letter queue. Either manually remove the publication from the stream queue, or correct the problem that is preventing the queue manager from writing the publication to the dead-letter queue.

CSQT857E

csect-name Unable to process control command message, reason=*mqrc (mqrc-text)*

Severity
8

Explanation

The Queued Pub/Sub Daemon failed to process a command message on the SYSTEM.BROKER.CONTROL.QUEUE. The queue manager was unable to write the command message to the dead-letter queue and was not permitted to discard the command message. The queue manager will temporarily stop the stream and will restart the stream and consequently retry the command message after a short interval. Other queue manager control commands cannot be processed until this command message has been processed successfully or removed from the control queue.

System programmer response

Investigate why the error has occurred and why the command message cannot be written to the dead-letter queue. Either, manually remove the command message from the stream queue, or correct the problem that is preventing the broker from writing the command message to the dead-letter queue.

CSQT858E

csect-name Unable to send publication to subscriber queue, queue *queue-name*, to queue manager *queue_manager_name*, reason=*mqrc* (*mqrc-text*)

Severity
8

Explanation

A failure has occurred sending a publication to subscriber queue *queue-name* at queue manager *queue_manager_name* for reason *mqrc*. The broker configuration options prevent it from recovering

from this failure by discarding the publication or by sending it to the dead-letter queue. Instead the queue manager will back out the unit of work under which the publication is being sent and retry the failing command message a fixed number of times. If the problem still persists, the queue manager will then attempt to recover by failing the command message with a negative reply message. If the issuer of the command did not request negative replies, the queue manager will either discard or send to the dead-letter queue the failing command message. If the queue manager configuration options prevent this, the queue manager will restart the affected stream, which will reprocess the failing command message again. This behavior will be repeated until such time as the failure is resolved. During this time the stream will be unable to process further publications or subscriptions.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form).

Usually the failure will be due to a transient resource problem, for example, the subscriber queue, or an intermediate transmission queue, becoming full. Use reason code *mqrc* to determine what remedial action is required. If the problem persists for a long time, you will notice the stream being continually restarted by the queue manager. Evidence of this occurring will be a large number of CSQT820E messages, indicating stream restart, being written to the Channel Initiator log. In such circumstances, manual intervention will be required to allow the queue manager to dispose of the failing publication. To do this, you will need to end the Queued Pub/Sub Daemon using the ALTER QMGR PSMODE(COMPAT), change the appropriate queue manager attributes; PSNPMSG, PSNPRES, PSSYNCPT, and restart it using ALTER QMGR PSMODE(ENABLED). This will allow the publication to be sent to the rest of the subscribers, while allowing the Queued Pub/Sub Daemon to discard or send to the dead-letter queue the publication that could not be sent.

CSQT859E

csect-name Queued Pub/Sub stream *stream-name* terminating, reason=*mqrc (mqrc-text)*

Severity

8

Explanation

The stream *stream-name* has run out of internal resources and will terminate with reason code *mqrc* (*mqrc-text* provides the MQRC in textual form). If the command in progress was being processed under syncpoint control, it will be backed out and retried when the stream is restarted by the queue manager. If the command was being processed out of syncpoint control, it will not be able to be retried when the stream is restarted.

System programmer response

This message should only be issued in very unusual circumstances. If this message is issued repeatedly for the same stream, and the stream is not especially large in terms of subscriptions, topics, and retained publications, save all generated diagnostic information and use either the IBM MQ Support site, or IBM Support Assistant (ISA) to see whether a solution is already available. If you are unable to find a match, contact IBM Support.

CSQT864E

csect-name Unable to put a reply message, queue *queue-name* queue manager(*qm-name*) MQCC=*mqcc* MQRC=*mqrc* (*mqrc-text*)

Severity

8

Explanation

While processing a publish/subscribe command, the queue manager could not send a reply message to the queue *queue-name* at the queue manager *qm-name* for MQRC=*mqrc*. The queue manager was also unable to write the message to the dead-letter queue. Since the command is being processed under syncpoint control, the queue manager will attempt to retry the command in the hope that the problem is only of a transient nature. If, after a set number of retries, the reply message still could

not be sent, the command message will be discarded if the report options allow it. If the command message cannot be discarded, the stream will be restarted, and processing of the command message recommenced.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form).

Use reason code *mqrc* to determine what remedial action is required. If the failure is due to a resource problem (for example, a queue being full), you might find that the problem has already cleared itself. If not, this message will be issued repeatedly each time the command is retried. In this case you are strongly advised to define a dead-letter queue to receive the reply message so that the Queued Pub/Sub Daemon can process other commands while the problem is being investigated. Check the application from which the command originated and ensure that it is specifying its reply-to queue correctly.

CSQT866E

csect-name Queued Pub/Sub command message discarded. Reason=*mqrc (mqrc-text)*

Severity
8

Explanation

The queue manager failed to process a publish/subscribe command message, which has now been discarded. The queue manager will begin to process new command messages again.

System programmer response

Look for previous error messages to indicate the problem with the command message. Correct the problem to prevent the failure from happening again.

CSQT875E

csect-name Unable to put message to the dead-letter-queue, reason=*mqrc (mqrc-text)* (DLH reason=*mqrc2 (mqrc2-text)*)

Severity
8

Explanation

The queue manager attempted to put a message to the dead-letter queue *queue-name* but the message could not be written to the dead-letter queue for reason *mqrc (mqrc-text* provides the MQRC in textual form). The message was being written to the dead-letter-queue with a reason of *mqrc2* (*mqrc2-text* provides the MQRC in textual form).

System programmer response

Determine why the message cannot be written to the dead-letter-queue. Also, if the message was not deliberately written to the dead-letter-queue, for example by a channel exit, determine why the message was written to the dead-letter-queue and resolve the problem that is preventing the message from being sent to its destination.

CSQT876E

csect-name Parent conflict detected in Pub/Sub hierarchy with queue manager *queue_manager_name*

Severity
8

Explanation

The queue manager *queue_manager_name* has been started, naming this queue manager as its parent. This queue manager has already named queue manager *queue_manager_name* as its parent. The queue manager will send an exception message to the queue manager *queue_manager_name* indicating that a conflict has been detected. The most likely reason for this message is that the queue

manager topology has been changed while distributed publish/subscribe communication messages were in transit (for example, on a transmission queue) and that a message relating to the previous queue manager topology has arrived at a queue manager in the new topology. This message might be accompanied by an informational FFST including details of the unexpected communication.

System programmer response

If the queue manager topology has changed and the queue manager named in the message no longer identifies this queue manager as its parent, this message can be ignored - for example, if the command ALTER QMGR PARENT(' ') was issued. If queue manager *queue_manager_name* has been defined as this queue manager's parent, and this queue manager has been defined as queue manager *queue_manager_name*'s parent, the ALTER QMGR command should be used to resolve the conflict by specifying the correct PARENT.

CSQT882E

csect-name Message written to the dead-letter queue, for reason=*mqrc (mqrc-text)*

Severity
8

Explanation

The queue manager has written a message to the dead-letter queue for reason *mqrc* (*mqrc-text* provides the MQRC in textual form). Note. After the first occurrence of this message for a stream, it will only be written periodically.

System programmer response

Determine why the message was written to the dead-letter queue, and resolve the problem that is preventing the message from being sent to its destination.

CSQT883E

csect-name Queued Pub/Sub state not recorded

Severity
0

Explanation

The Queued Pub/Sub state on stream *stream-name* not recorded while processing a publication outside of syncpoint. A nonpersistent publication has requested a change to either a retained message or a publisher registration. This publication is being processed outside of syncpoint because the queue manager has been configured with the queue manager attribute PSSYNCPT set to IFPER. A failure has occurred hardening either the publisher registration or the retained publication to the queue manager's local queue. All state changes attempted as a result of this publication will be backed-out. Processing of the publication will continue and the queue manager will attempt to deliver it to all subscribers.

System programmer response

Investigate why the failure occurred. It is probably due to a resource problem occurring on the queue manager. The most likely cause is 'queue full' on a queue. If your publications also carry state changes, you are advised to send them either as persistent publications or set the queue manager attribute PSSYNCPT to YES. In this way, they will be carried out under syncpoint and the queue manager can retry them in the event of a failure such as this.

CSQT884E

csect-name Queued Pub/Sub control queue is not a local queue

Severity
8

Explanation

The queue manager has detected that the queue SYSTEM.BROKER.CONTROL.QUEUE exists and is not a local queue. This makes the queue unsuitable for use as the control queue. The Pub/Sub Daemon task will terminate immediately.

System programmer response

Delete the definition of the existing queue and, if required, re-create the queue to be of type MQQT_LOCAL.

CSQT895I

csect-name Queued Pub/Sub Daemon detected missing retained messages

Severity
4

Explanation

The Queued Pub/Sub Daemon uses retained messages to communicate with other members of publish subscribe hierarchies.

The retained message was missing and has been republished.

System action

Retained messages seem to have been removed from the SYSTEM.RETAINED.PUB.QUEUE. The Queued Pub/Sub Daemon has attempted to recover by republishing retained messages.

System programmer response

If you are unaware of a reason why retained messages have been removed this might be a symptom of a more serious problem that requires further investigation.

CSQT899E

csect-name Unable to establish parent relationship to child queue manager *qmname*

Severity
8

Explanation

The queue manager is unable to establish the requested parent relationship to queue manager *qmname* because that queue manager is already a child.

System action

The existing child relationship to queue manager *qmname* remains unaffected.

System programmer response

To prevent this message being issued, the parent definition on the queue manager must be removed by issuing the **ALTER QMGR PARENT(' ')** MQSC command. To ensure that the required topology is established, review the existing parent definitions and update appropriately.

CSQT960I

csect-name Distributed Pub/Sub command processor stopped

Severity
0

Explanation

The distributed Pub/Sub command processor stopped. This may be for one of three reasons:

- The channel initiator is stopping.
- The channel initiator is starting and the queues used by the distributed Pub/Sub command processor have not been defined because distributed Pub/Sub command processor is not required.

- An error has occurred

System action

Processing continues, but distributed Pub/Sub is not available.

System programmer response

If an error has occurred, investigate the problem reported in the preceding messages.

CSQT961I

csect-name Distributed Pub/Sub publication processor stopped

Severity
0

Explanation

The distributed Pub/Sub publication processor stopped. This can be for one of three reasons:

- The channel initiator is stopping.
- The channel initiator is starting and the queues used by the distributed Pub/Sub command processor have not been defined because distributed Pub/Sub publication processor is not required.
- An error has occurred

System action

Processing continues, but distributed Pub/Sub is not available.

System programmer response

If an error has occurred, investigate the problem reported in the preceding messages.

CSQT962I

csect-name Distributed Pub/Sub proxy-subscription fan out processor stopped

Severity
0

Explanation

The distributed Pub/Sub proxy-subscription stopped. This can be for one of three reasons:

- The channel initiator is stopping.
- The channel initiator is starting and the queues used by the distributed pub/sub proxy-subscription fan out processor have not been defined because distributed pub/sub proxy-subscription fan out processor is not required.
- An error has occurred

System action

Processing continues, but distributed Pub/Sub is not available.

System programmer response

If an error has occurred, investigate the problem reported in the preceding messages.

CSQT963E

csect-name Queued pub/sub daemon unavailable

Severity
8

Explanation

The Distributed publish/subscribe process was unable to contact the Queued Pub/Sub Daemon. The problem will be reported in preceding messages.

System action

Hierarchical connections cannot be processed until the problem is rectified.

System programmer response

Investigate the problem reported in the preceding messages. When the Daemon becomes available, it might be necessary to issue the REFRESH QMGR TYPE(PROXYSUB) command to resynchronize subscriptions.

CSQT964I

csect-name Pub/Sub hierarchy relation connected, (queue manager *qmgr-name*)

Severity
0

Explanation

A publish/subscribe hierarchy connection has been established with child or parent queue manager *qmgr-name*.

CSQT965I

csect-name Pub/Sub hierarchy relation disconnected, (queue manager *qmgr-name*)

Severity
0

Explanation

A publish/subscribe hierarchy connection has ended with child or parent queue manager *qmgr-name*.

CSQT966E

csect-name A previous publication is being incorrectly processed again

Severity
8

Explanation

A publication, previously processed by this queue manager, has been received.

This is caused by an invalid configuration of a hierarchy and a pub/sub cluster.

System action

This message will not be re-published and will be processed according to the message's report options. Additional messages might be written if this publication is sent to the dead-letter queue.

System programmer response

Correct the configuration to remove the loop. Check the message properties in the dead-letter queue to determine the route taken.

CSQT967E

csect-name Unable to deliver proxy subscription to queue manager *queue_manager_name*, reason=*mqrc (mqrc-text)*

Severity
8

Explanation

Unable to deliver proxy subscription to queue manager *queue_manager_name*. Reason code: *mqrc* (*mqrc-text* provides the MQRC in textual form).

This might result in subscriptions not receiving publications from *queue_manager_name*.

System programmer response

Correct the configuration to allow proxy subscriptions to be delivered to *queue_manager_name*. When the problem has been resolved, it will be necessary to perform a **REFRESH QMGR TYPE(PROXYSUB)** to resynchronize subscriptions.

CSQT968I

csect-name Topic *topic-1* in cluster *cluster_name* is dependent on PROXYSUB(FORCE) of topic *topic-2*

Severity

4

Explanation

Topic object *topic-1* is defined in cluster *cluster_name*, and is below topic object *topic-2* in the topic tree. Topic object *topic-2* has been configured with **PROXYSUB(FORCE)** to generate a wildcard proxy subscription, so no further individual proxy subscriptions are sent for any subscriptions made below *topic-2* in the topic tree. However, *topic-2* is not in the same cluster as *topic-1*, and the wildcard proxy subscription is not sent to neighboring queue managers in the cluster in which *topic-1* is defined. Therefore, publications from cluster *cluster_name* might not be sent to subscriptions to *topic-1* on this queue manager.

System programmer response

If the behavior described in the explanation is intended no action is required. If not, alter the **PROXYSUB** attribute on topic *topic-1*, or *topic-2*, so both, or neither topics, are configured with the value **FORCE**.

CSQT971E

csect-name task failed to quiesce

Severity

8

Explanation

The indicated Distributed Publish/Subscribe task was requested to quiesce but failed to do so within the timeout interval.

There are four classes of task:

Distributed Pub/Sub Publish Task
Receives publications from remote queue managers in a Publish/Subscribe cluster and republishes into the local queue manager

Distributed Pub/Sub Command Task
Receives command messages from remote queue managers in a Publish/Subscribe cluster to create or cancel proxy subscriptions on behalf of remote queue managers.

Distributed Pub/Sub Fan Out Task
Sends command messages to remote queue managers in Publish/Subscribe clusters and Publish/Subscribe hierarchies in response to changes in the local queue manager state.

Distributed Pub/Sub Controller
Controls the starting and stopping of the Distributed Publish/Subscribe tasks during channel initiator startup and shutdown and also when enabling and disabling Publish/Subscribe.

System action

The Queued Pub/Sub Daemon will be forcibly closed.

System programmer response

Check the job log for additional messages, or an FFST™, that might explain why the task has failed to quiesce.

CSQT972E

csect-name Unable to put Distributed Pub/Sub fan-out request to *q-name*, reason=*mqrc (mqrc-text)*

Severity
8

Explanation

An attempt to put a subscription fan-out request on the distributed publish/subscribe fan-out request queue *q-name* failed with reason code *mqrc (mqrc-text* provides the MQRC in textual form).

CSQT973E

csect-name Distributed Pub/Sub subscribing inhibited, topic string *topic-string*, (queue manager *qm-name*)

Severity
8

Explanation

Topic *topic-string* has been disabled for subscribe. This prevents distributed publish/subscribe from creating a subscription on behalf of another queue manager *qm-name* within the topology.

CSQT974E

csect-name Distributed Pub/Sub publication inhibited, topic string *topic-string*

Severity
8

Explanation

Topic *topic-string* has been disabled for publish. This prevents distributed publish/subscribe from publishing a message received from another queue manager within the topology. This message will not be re-published and will be processed according to the report options in the message. Additional messages will be written if this publication is sent to the dead-letter queue.

CSQT975I

csect-name task has started

Severity
0

Explanation

The indicated Distributed Publish/Subscribe task has started. This message typically occurs during channel initiator startup, or when enabling Publish/Subscribe.

There are four classes of task:

Distributed Pub/Sub Publish Task
Receives publications from remote queue managers in a Publish/Subscribe cluster and republishes into the local queue manager

Distributed Pub/Sub Command Task
Receives command messages from remote queue managers in a Publish/Subscribe cluster to create or cancel proxy subscriptions on behalf of remote queue managers.

Distributed Pub/Sub Fan Out Task
Sends command messages to remote queue managers in Publish/Subscribe clusters and Publish/Subscribe hierarchies in response to changes in the local queue manager state.

Distributed Pub/Sub Controller
Controls the starting and stopping of the Distributed Publish/Subscribe tasks during channel initiator startup and shutdown, and also when enabling and disabling Publish/Subscribe.

System action

None.

System programmer response

None.

CSQT976I

csect-name task has stopped

Severity
0

Explanation

The indicated Distributed Publish/Subscribe task has stopped. This message typically occurs during channel initiator shutdown, or when disabling Publish/Subscribe.

There are four classes of task:

Distributed Pub/Sub Publish Task
Receives publications from remote queue managers in a Publish/Subscribe cluster and republishes into the local queue manager

Distributed Pub/Sub Command Task
Receives command messages from remote queue managers in a Publish/Subscribe cluster to create or cancel proxy subscriptions on behalf of remote queue managers.

Distributed Pub/Sub Fan Out Task
Sends command messages to remote queue managers in Publish/Subscribe clusters and Publish/Subscribe hierarchies in response to changes in the local queue manager state.

Distributed Pub/Sub Controller
Controls the starting and stopping of the Distributed Publish/Subscribe tasks during channel initiator startup and shutdown and also when enabling and disabling Publish/Subscribe.

System action

None.

System programmer response

None.

CSQT977I

csect-name Establishing Pub/Sub hierarchy relation, (queue manager *qmgr-name*)

Severity
0

Explanation

The queue manager is establishing a Publish/Subscribe hierarchy connection with a child or parent queue manager *qmgr-name*.

System action

None.

System programmer response

None.

CSQT978E

csect-name Unable to create/cancel proxy subscription, for queue manager *queue_manager_name*, topic string *topic-string*, reason=*mqrc* (*mqrc-text*)

Severity
8

Explanation

The Distributed Pub/Sub Command Task is unable to create or cancel a proxy subscription for queue manager *queue_manager_name* on topic *topic-string* for reason code *mqrc* (*mqrc-text* provides the MQRC in textual form).

A failure to create or cancel a proxy subscription will result in this queue manager not having a correct knowledge of subscriptions on other queue managers in the Publish/Subscribe topology. This may result in this queue manager not delivering publications to other queue managers.

System programmer response

Correct the cause of the indicated reason code.

Once the problem has been resolved it may be necessary to perform a REFRESH QMGR TYPE(PROXYSUB) command to resynchronise any subscriptions.

CSQT979E

csect-name Distributed Pub/Sub proxy subscription from qmgr-name rejected due to PSCLUS(DISABLED)

Explanation

A cluster subscription has been sent to this queue manager over a channel from qmgr-name but the queue manager attribute PSCLUS has been set to DISABLED, indicating that Publish/Subscribe activity is not expected between queue managers in this cluster.

System action

The proxy subscription request is ignored and no subscription is locally registered.

System programmer response

To enable publish/subscribe clustering, alter the PSCLUS attribute on all queue managers in the cluster to ENABLED. You may also need to issue **REFRESH CLUSTER** and **REFRESH QMGR** commands as detailed in the documentation for the PSCLUS attribute. If you are not using publish/subscribe clusters you should delete the clustered topic object, and ensure PSCLUS is DISABLED on all queue managers.

CSQT980I

csect-name Distributed Pub/Sub proxy subscription re-synchronization completed

Severity

0

Explanation

During restart processing the Distributed Pub/Sub process was unable to determine that the proxy subscription state was consistent so a re-synchronization with remote queue managers has been performed.

This is usually seen when a queue manager was not quiesced cleanly during its previous shutdown, or when the system was particularly busy at that time.

System action

Processing continues.

System programmer response

None.

CSQT981E

csect-name Distributed Pub/Sub disabled whilst in a Pub/Sub cluster

Severity

4

Explanation

This queue manager is a member of a Publish/Subscribe cluster but Publish/Subscribe has been disabled.

System action

Other queue managers within the Publish/Subscribe Cluster will continue to send publications and proxy subscriptions to this queue manager. They will accumulate on the Publish/Subscribe Cluster system queues and will not be processed until Publish/Subscribe is enabled. If these queues become full channel failure may occur, which will affect the operation of Publish/Subscribe on other queue managers in the cluster. This will also affect the delivery of other messages, unrelated to Publish/Subscribe, that are sent to this queue manager from other queue managers within the cluster.

System programmer response

Enable Publish/Subscribe by setting **PSMODE** to **ENABLED** or **COMPAT** with the **ALTER QMGR** command then the **REFRESH QMGR TYPE(PROXYSUB)** command should be issued to resynchronise subscriptions.

CSQT982E

csect-name Queued Pub/Sub disabled whilst in a Pub/Sub hierarchy

Severity
4

Explanation

This queue manager is a member of a Publish/Subscribe hierarchy but Queued Publish/Subscribe has been disabled.

System action

Any parent-child relations within the Publish/Subscribe hierarchy will continue to send publications and proxy subscriptions to this queue manager. They will accumulate on the Queued Publish/Subscribe system queues and will not be processed until Queued Publish/Subscribe is enabled. If the Queued Publish/Subscribe system queues become full channel failure may occur, which will affect the operation of Publish/Subscribe on parent-child relations sending messages to this queue manager. This will also affect the delivery of other messages, unrelated to Publish/Subscribe, that are to be delivered using the same channels.

System programmer response

Enable Queued Publish/Subscribe by setting **PSMODE** to **ENABLED** with the **ALTER QMGR** command. Once Queued Publish/Subscribe has been restarted, use the **DISPLAY PUBSUB ALL** command to confirm this has completed, the **REFRESH QMGR TYPE(PROXYSUB)** command must be issued to resynchronize

CSQT983E

csect-name task failed, reason *mqrc (mqrc-text)*, retry in *n* minutes

Severity
4

Explanation

The *task* encountered a problem and will retry the command in *n* minutes. Earlier messages might have been issued in the queue manager or system error logs providing additional detail.

This message might be issued by a number of tasks:

Distributed Pub/Sub Publish Task
Other queue managers within the cluster will continue to send publications to this queue manager. The publications will accumulate on the Publish/Subscribe Cluster system queue (SYSTEM.INTER.QMGR.PUBS) and will not be processed until the problem is resolved. If these queues become full channel failure might occur, which will affect the operation of Publish/

Subscribe on other queue managers in the cluster. This will also affect the delivery of other messages, unrelated to Publish/Subscribe, that are sent to this queue manager from other queue managers in the cluster.

Distributed Pub/Sub Command Task
Other queue managers within the cluster will continue to send proxy subscriptions to this queue manager. Subscriptions will accumulate on the Publish/Subscribe Cluster system queue (SYSTEM.INTER.QMGR.CONTROL) and will not be processed until the problem is resolved. Other queue managers will not receive publications from this queue manager on topics for which proxy subscriptions have yet to be processed. If the Publish/Subscribe Cluster system queue becomes full channel failure might occur, which will affect the operation of Publish/Subscribe on other queue managers in the cluster. This will also affect the delivery of other messages, unrelated to Publish/Subscribe, that are sent to this queue manager from other queue managers in the cluster.

Distributed Pub/Sub Fan Out Task
This task will not send proxy subscription messages to other queue managers in a Publish/Subscribe Cluster or parent-child relations within a Publish/Subscribe hierarchy until the problem is rectified. On topics for which proxy subscriptions have yet to be sent, this queue manager will not receive publications from other queue managers in a Publish/Subscribe Cluster, or parent-child relations in a Publish/Subscribe hierarchy.

System action

None

System programmer response

If possible, rectify the identified problem, or contact IBM Support.

When the problem has been rectified wait for *task* to retry the command.

CSQT984E

csect-name task has encountered *n* occurrences of reason *mqrc (mqrc-text)* while attempting to process a message.

Severity
4

Explanation

The *task* is currently unable to process a message due to reason *mqrc* (*mqrc-text* provides the MQRC in textual form). The task has encountered this *n* times; it will continue to retry the command until the problem has been rectified.

This message might be issued by a number of tasks:

Distributed Pub/Sub Publish Task
Other queue managers within the cluster will continue to send publications to this queue manager. Publications will accumulate on the Publish/Subscribe Cluster system queue (SYSTEM.INTER.QMGR.PUBS) and will not be processed until the problem is resolved. If these queues become full channel failure might occur, which will affect the operation of Publish/Subscribe on other queue managers in the cluster. This will also affect the delivery of other messages, unrelated to Publish/Subscribe, that are sent to this queue manager from other queue managers within the cluster.

Distributed Pub/Sub Command Task
Other queue managers within the cluster will continue to send proxy subscriptions to this queue manager. Subscriptions will accumulate on the Publish/Subscribe Cluster system queue (SYSTEM.INTER.QMGR.CONTROL) and will not be processed until the problem is resolved. Other queue managers will not receive publications from this queue manager on topics for which proxy subscriptions have yet to be processed. If the Publish/subscriber Cluster system queue becomes full channel failure might occur, which will affect the operation of Publish/Subscribe on other queue managers in the cluster. This will also affect the delivery of other messages, unrelated to Publish/Subscribe, that are sent to this queue manager from other queue managers within the cluster.

Distributed Pub/Sub Fan Out Task
This task will not send proxy subscription messages to other queue managers in a Publish/Subscribe Cluster or parent-child relations within a Publish/Subscribe hierarchy until the problem is rectified. On topics for which proxy subscriptions have yet to be sent this queue manager will not receive publications from other queue managers in a Publish/Subscribe Cluster or parent-child relations within a Publish/Subscribe hierarchy.

System action

None

System programmer response

If possible, rectify the identified problem, or contact IBM Support.

When the problem has been rectified wait for *task* to retry the command.

CSQT987E

csect-name task failed due to reason *mqrc (mqrc-text)* Retry in *n* minutes

Severity
4

Explanation

The *task* encountered a problem. Earlier messages might have been issued in the queue manager or system error logs providing additional detail. The task will retry the command in *n* minutes.

Other queue managers within the cluster will continue to send proxy subscriptions to this queue manager. They will accumulate on the Publish/Subscribe cluster system queue and will not be processed until the problem is resolved.

Other queue managers will not receive publications from this queue manager on topics for which proxy subscriptions have yet to be processed.

If the Publish/subscriber cluster system queue becomes full, channel failure might occur, which will affect the operation of Publish/Subscribe on other queue managers in the cluster. This will also affect the delivery of other messages, unrelated to Publish/Subscribe, that are sent to this queue manager from other queue managers within the cluster.

System action

None

System programmer response

If possible, rectify the identified problem, or contact IBM Support.

When the problem has been rectified wait for *task* to retry the command.

CSQT988E

csect-name task failed due to reason *mqrc (mqrc-text)* Retry in *n* minutes

Severity
4

Explanation

The *task* encountered a problem. Earlier messages might have been issued in the queue manager or system error logs providing additional detail. The task will retry the command in *n* minutes.

This task will not send proxy subscription messages to other queue managers in a Publish/Subscribe cluster or parent-child relations within a Publish/Subscribe hierarchy until the problem is rectified.

On topics for which proxy subscriptions have yet to be sent, this queue manager will not receive publications from other queue managers in a Publish/Subscribe cluster or parent-child relations within a Publish/Subscribe hierarchy.

System action

None

System programmer response

If possible, rectify the identified problem, or contact IBM Support.

When the problem has been rectified wait for *task* to retry the command.

CSQT989E

csect-name task has encountered *n* occurrences of reason *mqrc (mqrc-text)* while attempting to process a message.

Severity
4

Explanation

The *task* is currently unable to process a message due to reason *mqrc (mqrc-text)*. Note that (*(mqrc-text)* provides the MQRC in textual form).

The task has encountered this *n* times. The task continues to retry the command until the problem has been rectified.

Other queue managers within the cluster continue to send proxy subscriptions to this queue manager. The subscriptions will accumulate on the Publish/Subscribe cluster system queue and will not be processed until the problem is resolved.

Other queue managers will not receive publications from this queue manager on topics for which proxy subscriptions have yet to be processed.

If the Publish/subscriber cluster system queue becomes full, channel failure might occur, which will affect the operation of Publish/Subscribe on other queue managers in the cluster.

This will also affect the delivery of other messages, unrelated to Publish/Subscribe, that are sent to this queue manager from other queue managers within the cluster.

System action

None

System programmer response

If possible, rectify the identified problem, or contact IBM Support.

When the problem has been rectified wait for *task* to retry the command.

CSQT990E

csect-name task has encountered *n* occurrences of reason *mqrc (mqrc-text)* while attempting to process a message.

Severity
4

Explanation

The *task* is currently unable to process a message due to reason *mqrc (mqrc-text)*. Note that (*(mqrc-text)* provides the MQRC in textual form).

The task has encountered this *n* times. The task continues to retry the command until the problem has been rectified.

This task will not send proxy subscription messages to other queue managers in a Publish/Subsccribe Cluster or parent-child relations within a Publish/Subscribe hierarchy until the problem is rectified.

On topics for which proxy subscriptions have yet to be sent this queue manager will not receive publications from other queue managers in a Publish/Subscribe cluster or parent-child relations within a Publish/Subscribe hierarchy.

System action

None

System programmer response

If possible, rectify the identified problem, or contact IBM Support.

When the problem has been rectified wait for *task* to retry the command.

CSQT991I

csect-name task has recovered from previous error condition

Severity
0

Explanation

The *task* has recovered from the previously reported error condition.

System action

Processing continues.

System programmer response

None.

CSQT992E

csect-name task has written a message to the dead-letter queue, reason *mqrc (mqrc-text)*

Severity
8

Explanation

The *task* has written a message to the dead-letter queue due to reason *mqrc* (*mqrc-text* provides the MQRC in textual form).

If *task* is the Distributed Pub/Sub Command Task, other queue managers will not receive publications from this queue manager on any topics for which this message is requesting proxy subscriptions be created.

If *task* is the Distributed Pub/Sub Fan Out Task, this queue manager will not receive publications from other queue managers on any topics for which this is requesting a proxy subscription be created.

System programmer response

Determine why the message was written to the dead-letter queue, and resolve the problem that is preventing the message from being sent to its destination.

If *task* is the Distributed Pub/Sub Command Task, or the Distributed Pub/Sub Fan Out Task, it may be necessary to issue the **REFRESH QMGR TYPE(PROXYSUB)** command when the problem has been resolved to resynchronize the subscription state with other queue managers.

CSQT996E

csect-name Creation of proxy subscription failed on queue manager *qmgr-name*, cluster *cluster_name*, topic string *topic-string*, reason=*mqrc (mqrc-text)*

Severity
8

Explanation

The proxy subscription in publish/subscribe cluster *cluster_name* on topic *topic-string* could not be created on queue manager *qmgr-name* due to reason *mqrc* (*mqrc-text* provides the MQRC in textual form). The failure to create the proxy subscription will prevent publications made on queue manager *qmgr-name* to topic *topic-string* being delivered to subscriptions on this queue manager. If this queue manager is also participating as a member of a publish/subscribe hierarchy any subscriptions to topic

topic-string on other members of the publish/subscribe hierarchy will not receive publications from queue manager *qmgr-name*.

System programmer response

Correct the cause of the indicated reason code on queue manager *qmgr-name*. When the problem has been resolved issue the **REFRESH QMGR TYPE(PROXYSUB)** command on the remote queue manager to resynchronize the subscription state with other queue managers.

CSQT997E

csect-name Cancellation of proxy subscription failed on queue manager *qmgr-name*, cluster *cluster_name*, topic string *topic-string*, reason=*mqrc (mqrc-text)*

Severity

8

Explanation

The proxy subscription in publish/subscribe cluster *cluster_name* on topic *topic-string* could not be canceled on queue manager *qmgr-name* due to reason *mqrc* (*mqrc-text* provides the MQRC in textual form). The failure to cancel the proxy subscription will result in publications made on queue manager *qmgr-name* to topic *topic-string* to continue being delivered to this queue manager.

System programmer response

Correct the cause of the indicated reason code on queue manager *qmgr-name*. When the problem has been resolved issue the **REFRESH QMGR TYPE(PROXYSUB)** command on the remote queue manager to resynchronize the subscription state with other queue managers.

CSQT998E

csect-name Proxy subscription re-synchronization failed on queue manager *qmgr-name*, cluster *cluster_name*, reason=*mqrc (mqrc-text)*

Severity

8

Explanation

The request to resynchronize the subscription state with other queue managers in publish/subscribe cluster *cluster_name* failed on queue manager *qmgr-name* due to reason *mqrc* (*mqrc-text* provides the MQRC in textual form). There might be topic strings for which proxy subscriptions have not been created. Publications made on queue manager *qmgr-name* to those topics will not be delivered to subscriptions on this queue manager. If this queue manager is also participating as a member of a publish/subscribe hierarchy any subscriptions to those topics on other members of the publish/subscribe hierarchy will not receive publications from queue manager *qmgr-name*. There might also be topic strings for which proxy subscriptions have not been canceled on queue manager *qmgr-name*. Any publications made on that queue manager will continue to be delivered to this queue manager.

System programmer response

Correct the cause of the indicated reason code on queue manager *qmgr-name*. When the problem has been resolved issue the **REFRESH QMGR TYPE(PROXYSUB)** command on the remote queue manager to resynchronize the subscription state with other queue managers.

CSQT999E

csect-name task has encountered a message that is not valid on queue *queue*

Severity

4

Explanation

The queue *queue* is for exclusive use by the internal queue manager task *task*, and is used to maintain a distributed publish/subscribe topology. The task has encountered a message on the queue that is not valid

System action

The message is processed according to its report options. Additional console messages might be output if the message is put to the dead-letter queue.

An informational FFST, including details of the message that is not valid, might also be generated.

System programmer response

Ensure no applications put messages directly to the named queue, and ensure message exits do not alter system messages put to the queue. If the problem persists contact IBM Support.

z/OS *Utilities messages (CSQU...)*

CSQU000I
csect-name IBM MQ for z/OS V*n*

Explanation

This is part of the header to the report issued by the utility program.

CSQU001I
csect-name Queue Manager Utility - *date time*

Explanation

This is part of the header to the report issued by the utility program.

System action

The message is followed by a copy of the function statements from the SYSIN data set.

CSQU002E
Unable to get storage of size *n* bytes, return code=*ret-code*

Explanation

An attempt to obtain some storage failed.

System action

The function is terminated, and any queue updates are backed out.

System programmer response

If you encounter this error when submitting JCL to run CSQUTIL functions, make sure that you have defined an adequate value for the **REGION** size parameter or set the **REGION** size to `0M` in the JCL. For example:

```
//SCOPY EXEC PGM=CSQUTIL,REGION=0M //STEPLIB DD DISP=SHR,DSN=thlqual.SCSQANLE
 //DD DISP=SHR,DSN=thlqual.SCSQAUTH ...
```

For more details about setting the **REGION** parameter, see Copying queues into a data set while the queue manager is running (COPY).

See STORAGE or GETMAIN requests for information about the return codes, in the appropriate volume of the *z/OS MVS Programming: Assembler Services Reference* manual.

CSQU003E
Unable to free storage at *address*, return code=*ret-code*

Explanation

An attempt to release storage at address *address* back to the system failed.

System action

The program usually ignores the error and continues with its function.

System programmer response

See STORAGE or FREEMAIN requests for information about the return codes, in the appropriate volume of the *z/OS MVS Programming: Assembler Services Reference* manual.

CSQU005I
COMMIT successfully completed

Explanation

An MQCMIT call returned a completion code of MQCC_OK.

CSQU006I
BACKOUT successfully completed

Explanation

An MQBACK call returned a completion code of MQCC_OK.

System action

The function is terminated.

System programmer response

Investigate the error that caused the backout to be done.

CSQU007E
MQCMIT failed. MQCC=*mqcc* MQRC=*mqrc (mqrc-text)*

Explanation

The utility program was unable to commit the last set of changes.

System action

The updates are backed out, and the function is terminated.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form). Resubmit the job if required.

CSQU008E
MQBACK failed. MQCC=*mqcc* MQRC=*mqrc (mqrc-text)*

Explanation

The utility program was unable to back out the last set of changes.

System action

None, the function is already being terminated because of the error that led to attempting the backout.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form). Resubmit the job if required.

CSQU009E
MQCONN failed for *conn-id*. MQCC=*mqcc* MQRC=*mqrc (mqrc-text)*

Explanation

An attempt to connect to a queue manager or queue sharing group named *conn-id* was unsuccessful.

System action

The requested function is not performed.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form). Resubmit the job if required.

CSQU010E
MQDISC failed for *conn-id*. MQCC=*mqcc* MQRC=*mqrc (mqrc-text)*

Explanation

An attempt to disconnect from a queue manager or queue sharing group named *conn-id* was unsuccessful.

System action

The utility program terminates. (This is not an error, because the disconnection request is the last function that the utility program processes.)

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form).

CSQU011I
Commands from CSQINPX - *date time*

Explanation

This follows message CSQU000I as part of the header to the messages that indicate the progress of the utility program.

It is produced when the utility is invoked by distributed queuing to handle the CSQINPX data set.

CSQU012I
Initialization command handling completed

Explanation

The initialization command handler, which processes the CSQINPX command data set, completed successfully.

CSQU013E
Initialization command handling failed, RC=*return-code*

Explanation

The initialization command handler, which processes the CSQINPX command data set, did not complete successfully. *return-code* shows the type of error:

00000008
Some or all of the commands were not processed.

0000000C
Severe error; this is most likely because the CSQINPX or CSQOUTX data sets are defined erroneously.

System action

The initialization command handler ends, but the channel initiator continues.

System programmer response

Refer to the CSQOUTX data set and to the preceding messages for more information about the error.

For information about the initialization command handler and the CSQINPX or CSQOUTX data sets, see Initialization and configuration files. For information about the COMMAND statement, see Issuing commands to IBM MQ (COMMAND).

CSQU020E
Unable to OPEN *ddname* data set

Explanation

The program was unable to open data set *ddname*.

System action

If the SYSPRINT or SYSIN data sets cannot be opened, the utility program terminates. For other data sets, the function requesting them is not performed.

System programmer response

Examine the error message that was sent to the job log to determine the reason for the error. Check that the data set was correctly specified. Ensure that the data set record length is valid.

CSQU021E
Data set *ddname* does not have a record format of VBS

Explanation

The program opened the data set *ddname*, but the data set did not have a record format of VBS.

System action

If the LOAD input data set cannot be opened, the utility program terminates.

System programmer response

Examine the error message that was sent to the job log to determine the reason for the error. Check that the data set was correctly specified and is of the correct record format.

CSQU023E
Unable to CLOSE *ddname* data set

Explanation

The input data set *ddname* is still open after a request was made to close it.

System action

The program continues with its termination procedures.

System programmer response

Examine the error message that was sent to the job log to determine the reason for the error. Check that the data set was correctly specified.

CSQU030E
csect-name Page *nn* in data set *ddname* is invalid

Explanation

The utility program encountered a page that is invalid in the page set data set *ddname*. If the page number is 0, it might be that the data set is not the page set that is implied by *ddname*.

System action

The function is terminated.

System programmer response

Check that the page set has not been corrupted, and that the page set number corresponds to the DDname.

CSQU031E
Queue *q-name* with disposition QMGR or COPY does not exist

Explanation

The specified queue does not exist with disposition QMGR or COPY. (There might be such a queue with disposition SHARED, but the SCOPY function does not operate on shared queues.)

System action

The function is terminated.

System programmer response

Check the queue name that was specified.

CSQU032E
Page set *psid* is invalid

Explanation

The utility program encountered a page set that is invalid. The page set is in an inconsistent state and so the stand-alone utility functions cannot process it.

System action

The function is terminated.

System programmer response

This might be the result of performing a fuzzy backup (as described in How to back up and recover page sets) or because the queue manager terminated abnormally. Restart the queue manager and then terminate it normally.

CSQU036E
Utility not available - restricted functionality

Explanation

The utility cannot operate because the installation and customization options chosen for IBM MQ do not allow all functions to be used.

System action

The utility is terminated.

CSQU037I
function has been stabilized with *version* function

Explanation

The utility function identified by *function* has been stabilized with the functional capabilities of version *version*.

System action

Processing continues and additional messages might be output providing further information.

System programmer response

Review the use of the utility function.

CSQU038I
Use runmqsc -n on your client machine for client channel definitions from IBM MQ 8.0.

Explanation

The MAKECLNT utility function of CSQUTIL, that generates a client channel definition table (CCDT), has been stabilized.

From IBM MQ 8.0, use the runmqsc utility on the client machine to generate the CCDT instead.

CSQU040E
Unable to GET from *ddname* data set

Explanation

The program was unable to read a record from the *ddname* data set.

System action

The function is terminated, and any queue updates are backed out.

System programmer response

Examine the error message that was sent to the job log to determine the reason for the error. Check that the data set was correctly specified.

CSQU043E
Unable to PUT to *ddname* data set

Explanation

The program was unable to write the next record to the *ddname* data set. Either the data set was not opened, or there was a QSAM error.

System action

The function is terminated, and any queue updates are backed out.

System programmer response

Examine the error message that was sent to the job log to determine the reason for the error. Check that the data set was correctly specified.

CSQU044I
Commands cannot be made for queue managers other than the target, *qmgr-name*

Explanation

Some of the DISPLAY object commands for the COMMAND function with MAKEDEF, MAKEREP, MAKEALT, or MAKEDEL used the CMDSCOPE option, and so information about objects for queue managers other than the target queue manager *qmgr-name* was received. Commands are not generated for such objects.

System programmer response

Avoid using CMDSCOPE in conjunction with the MAKEDEF, MAKEREP, MAKEALT, or MAKEDEL options. Use a separate COMMAND function for each target queue manager, with separate data sets for each set of generated commands.

CSQU045I
n data records read

Explanation

This indicates how many data records were read from the input data set specified by the DATA keyword for the current function.

CSQU046I
Making client channel definitions in *ddname* data set using CCSID *ccsid*

Explanation

This indicates that the COMMAND function will build client channel definitions in data set *ddname*, and that the data will have a coded character set identifier of *ccsid*.

CSQU047E
Unable to convert data for client channel definitions. MQCC=*mqcc* MQRC=*mqrc (mqrc-text)*

Explanation

When building a client channel definition file, data for a channel or authentication information object could not be converted from the character set used by the queue manager to that requested by the CCSID keyword.

System action

The channel or authentication information definition is not built.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form). Resubmit the job if required.

CSQU048I
n authentication objects included, *m* excluded

Explanation

This indicates, for the current function, how many sets of authentication information were included in the client channel definition file, and how many were excluded. Authentication information may be excluded because:

- the LDAPUSER and LDAPPWD attributes are not blank
- there are too many sets of information
- there was a data conversion error.

System programmer response

If some information was excluded, check that the authentication information objects were selected correctly.

CSQU049I
n client channel definitions made

Explanation

This indicates how many client channel definitions were made by the current function.

CSQU050E
Command of length *length* is too long. Command rejected

Explanation

In the COMMAND function, the assembled command had more than 32 762 characters.

System action

The command is ignored, and no more commands are processed.

System programmer response

Check that the command is correctly formed according to the concatenation rules

CSQU051E
Command responses not received after *n* seconds

Explanation

In the COMMAND function, get processing for a response was timed out whilst more responses were expected.

System action

The next command will be processed, unless there have been too many timeouts.

System programmer response

Increase the value of RESPTIME, especially if the command is being sent to a remote queue manager, and check the remote queue definitions.

If the problem persists, check the definitions of the system-command input queue and the system-command reply queue; ensure that they are enabled for MQGET and MQPUT. If the definitions are correct, stop and restart the command server.

CSQU052E
Too many timeouts

Explanation

In the COMMAND function, get processing for a response timed out four times.

System action

No more commands are processed.

System programmer response

See message CSQU051E.

CSQU053E
DISPLAY command response not recognized

Explanation

In the COMMAND function, the responses to a DISPLAY command were not as expected.

System action

The DISPLAY command response is shown as is, rather than being formatted. The next command is processed.

System programmer response

Check the load libraries used are consistent with the queue manager being used.

Contact IBM Support to report the problem.

CSQU054I
Executing function for object type *objtyp*

Explanation

The utility program is executing function *function* to process objects of the type indicated.

CSQU055I
Target queue manager is *qmgr-name*

Explanation

This indicates which queue manager your commands are directed to.

CSQU056I
Making commands in *ddname* data set

Explanation

This indicates that commands for the COMMAND function with MAKEDEF, MAKEREP, MAKEALT, or MAKEDEL, or for the SDEFS function will be built in data set *ddname*.

CSQU057I
n commands read

Explanation

This indicates how many commands were read from the command input data set by the current function.

CSQU058I
n commands issued and responses received, *m* failed

Explanation

This indicates, for the current function, how many commands were sent and produced responses, and how many of these did not execute successfully.

CSQU059I
n cmd commands made

Explanation

This indicates how many commands (called *cmd*) were made for the current function.

CSQU060E

Incorrect data length for message *msg-no*. *act-length* bytes found, *exp-length* bytes expected

Severity
8

Explanation

In the LOAD or SLOAD function, when attempting to read the record for message number *msg-no* for the queue being processed, the actual record length was found to be different to the expected record length.

System action

Processing for the command is terminated.

System programmer response

Check that the data set was created by the COPY function.

CSQU061E
An error occurred accessing the *in-ddname* data set for message *msg-no*. Reason=*reason-code*

Explanation

When executing the LOAD, SLOAD or ANALYZE function and attempting to read message *msg-no* for queue being processed, an error was detected. The reason code specifies the specific error, as follows:

4
First record in the data set does not identify a queue

8
Unexpected end-of-file

12
Unknown record type

System action

Processing for the command is terminated.

System programmer response

Check that the data set was created by the COPY function, and is not corrupted.

CSQU062E
Incorrect format data record

Explanation

In the LOAD function, the utility program encountered a record that it does not recognize while reading from the input data set.

System action

The function is terminated, and any queue updates are backed out.

System programmer response

Check that the data set was created by the COPY function, and is not corrupted.

CSQU063E

The *in-ddname* data set is empty

Severity
8

Explanation

When executing the LOAD, SLOAD or ANALYZE function, the input data set (DDname *in-ddname*) was empty.

System action

Processing for the command is terminated.

System programmer response

Check that the data set was successfully created by the COPY function.

CSQU070I
Command processing stopped

Explanation

In the COMMAND function, with FAILURE(STOP) specified, a command did not execute successfully.

System action

No more commands are processed.

CSQU071E
Incomplete command

Explanation

In the COMMAND function, end of data on the input data set was reached before the building of a command was complete.

System action

The command is ignored. There are no more commands to process.

System programmer response

Check that the command is correctly formed according to the concatenation rules.

CSQU080E
MQCLOSE failed for queue *q-name*. MQCC=*mqcc* MQRC=*mqrc (mqrc-text)*

Explanation

The MQCLOSE call for *q-name* was unsuccessful. If this is for the system-command input queue when using the COMMAND function, message CSQU055I follows showing the target queue manager that was being used.

System action

The function is terminated.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form). Resubmit the job if required.

CSQU082E
MQGET failed for queue *q-name*. MQCC=*mqcc* MQRC=*mqrc (mqrc-text)*

Explanation

The MQGET call for *q-name* was unsuccessful.

System action

The function is terminated, and any queue updates are backed out.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form). Resubmit the job if required.

CSQU083E
MQOPEN failed for queue *q-name*. MQCC=*mqcc* MQRC=*mqrc (mqrc-text)*

Explanation

The MQOPEN call for *q-name* was unsuccessful. If the queue is a model queue, the requested dynamic queue name is appended in parentheses. If this is for the system-command input queue

when using the COMMAND function, message CSQU055I follows showing the target queue manager that was being used.

System action

The function is terminated, and all queue updates are backed out.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form). Resubmit the job if required.

CSQU085E
MQPUT failed for queue *q-name*. MQCC=*mqcc* MQRC=*mqrc (mqrc-text)*

Explanation

The MQPUT call for *q-name* was unsuccessful. If this is for the system-command input queue when using the COMMAND function, message CSQU055I follows showing the target queue manager that was being used.

System action

The function is terminated, and all queue updates are backed out.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form). Resubmit the job if required.

CSQU087I
MAXUMSGS reached. A syncpoint has been forced

Explanation

Because MAXUMSGS was reached, a syncpoint was taken which commits the queue changes made so far.

System action

The function continues, but no further functions will be processed.

System programmer response

None, unless the function fails for some reason after this message. In that case, note that some queue changes will have been committed, and you should make appropriate adjustments before rerunning the job.

CSQU090E
OPEN failed for *ddname* data set. VSAM return code=*rc* reason code=*reason*

Explanation

The utility program received a VSAM OPEN error for the page set it was attempting to process (pointed to by *ddname*).

System action

The page set is not processed.

System programmer response

See the *z/OS DFSMS Macro Instructions for Data Sets* manual for information about the return and reason codes from VSAM. If necessary, resubmit the job.

CSQU091E
ddname data set is non-empty. Page set not formatted

Explanation

Data set *ddname* was opened, but it is not empty.

System action

The page set is not formatted.

System programmer response

Ensure that the data sets specified are empty, and resubmit the job if necessary.

CSQU092I
function completed for *ddname* data set

Explanation

Processing of *ddname* data set for function *function* has completed.

System action

Processing continues with the next page set.

CSQU093E
PUT failed for *ddname* data set. VSAM return code=*rc* reason code=*code*

Explanation

The utility program received a VSAM PUT error for the page set it was attempting to process (pointed to by *ddname*).

System action

Processing for the page set is terminated, and the function continues with the next page set.

System programmer response

See the *z/OS DFSMS Macro Instructions for Data Sets* manual for information about the return and reason codes from VSAM. If necessary, resubmit the job.

CSQU094E
CLOSE failed for *ddname* data set. VSAM return code=*rc* reason code=*reason*

Explanation

The utility program received a VSAM CLOSE error for the page set it was attempting to process (pointed to by *ddname*).

System action

Processing for the page set is terminated, and the function continues with the next page set.

System programmer response

See the *z/OS DFSMS Macro Instructions for Data Sets* manual for information about the return and reason codes from VSAM. If necessary, resubmit the job.

CSQU095E
No page sets identified. *function* terminated

Explanation

A request to format or reset a page set was unsuccessful because there were no page set data sets with DD names in the range CSQP0000 through CSQP0099.

System action

Processing is terminated.

System programmer response

Add DD statements for the required page set data sets, and resubmit the job.

CSQU096I
csect-name FORMAT TYPE(*type*) skipped for page set *psid*

Explanation

The *type* of formatting for page set *psid* has been skipped because it is not allowed.

FORMAT TYPE(REINTRODUCE) is never allowed for page set zero. For other values, an earlier message will be issued which explains the problem.

System action

This page set is skipped, but FORMAT processing continues with the next page set.

System programmer response

Use the START QMGR BACKMIG command to migrate page set zero if required, and if it is allowed. Investigate earlier messages if necessary.

CSQU097E
csect-name FORMAT TYPE(type) TARGVER(*target-vrm*) failed for page set *psid*. Page set last used at *old-vrm*

Explanation

An attempt was made to use the FORMAT command to reintroduce the page set *psid* at version *target-vrm*. The page set was last used at version *old-vrm* and this migration is not supported by the utility. See FORMAT for a description of supported conversions.

System action

This page set is skipped, but FORMAT processing continues with the next page set.

System programmer response

Use the correct version of the utility if that is supported. If not, it might be possible to add the page set with FORMAT TYPE(NEW) or FORMAT TYPE(RECOVER) using the *target-vrm* version of the utility.

CSQU100E
ddname DD statement missing

Explanation

Data set *ddname* does not have a DD statement in the JCL.

System action

The utility is terminated.

System programmer response

Add the required statement to the JCL, and resubmit the job.

CSQU101E
DD statement missing for page set *psid*

Explanation

A page set is referenced, but there is no DD statement for it in the JCL. The DD name required is CSQP00*nn*, where *nn* is the page set number.

System action

The utility is terminated.

System programmer response

Add the required statement to the JCL, and resubmit the job.

CSQU102E
No functions requested

Explanation

There are no function statements in the SYSIN data set.

System action

The utility is terminated.

CSQU103E
Either keyword *keyword1* or *keyword2* must be specified

Explanation

The statement syntax is incorrect because it requires that one of the keywords *keyword1* or *keyword2* be specified, but not both.

System action

The utility is terminated.

System programmer response

See MQSC commands for information about the correct syntax required for the statement, then resubmit the job.

CSQU104E
Invalid value *value* for keyword *keyword*

Explanation

The statement syntax is incorrect because the value given for keyword *keyword* is not valid.

System action

The utility is terminated.

System programmer response

See MQSC commands for information about the correct syntax required for the statement, and resubmit the job.

CSQU105E
Incompatible keywords or values for function *function*

Explanation

The statement syntax is incorrect because a keyword or its value that is specified conflicts with another keyword or its value.

System action

The utility is terminated.

System programmer response

See MQSC commands for information about the correct syntax required for the statement, and resubmit the job.

CSQU106E
Invalid function *function*

Explanation

The statement syntax is incorrect because the function *function* is not recognized.

System action

The utility is terminated.

System programmer response

See MQSC commands for a list of valid functions, and resubmit the job.

CSQU107E
Invalid *function* statement syntax

Explanation

The syntax of the *function* statement is incorrect:

- there are too many keywords or values
- required keywords are missing
- it cannot be parsed.

System action

The utility is terminated.

System programmer response

See MQSC commands for information about the correct syntax required for the statement, and resubmit the job.

CSQU108E
Value missing for keyword *keyword*

Explanation

Keyword *keyword* should be followed by a value, but the value is missing.

System action

The utility is terminated.

System programmer response

See MQSC commands for information about the correct syntax required for the statement, and resubmit the job.

CSQU109E
Value not allowed for keyword *keyword*

Explanation

Keyword *keyword* should not be followed by a value, but a value is specified.

System action

The utility is terminated.

System programmer response

See Configuring z/OS for information about the correct syntax required for the statement, and resubmit the job.

CSQU110E
Required keyword missing for keyword *keyword*

Explanation

The statement syntax is incorrect because keyword *keyword* can be specified only if some other keyword is also specified, but that other keyword is missing.

System action

The utility is terminated.

System programmer response

See MQSC commands for information about the correct syntax required for the statement, then resubmit the job.

CSQU111E
Invalid keyword *keyword* for function *function*

Explanation

The statement syntax is incorrect because the keyword *keyword* is not valid for function *function*.

System action

The utility is terminated.

System programmer response

See MQSC commands for information about the correct syntax required for the statement, and resubmit the job.

CSQU112E
Incomplete statement

Explanation

End of data on the input data set was reached before the building of a statement was complete.

System action

The utility is terminated.

System programmer response

Check that the statement is correctly formed according to the concatenation rules.

CSQU113E
Too many statement continuations

Explanation

The statement has more than 10 continuations.

System action

The utility is terminated.

System programmer response

Check that the statement is correctly formed according to the concatenation rules.

CSQU114E
Keyword *keyword* repeated

Explanation

The statement syntax is incorrect because a keyword is repeated.

System action

The utility program is terminated.

System programmer response

Check the syntax in the input data set. See CSQUTIL for further information about the utility program.

CSQU115E
Unable to find queues for page set *psid* - command responses not received

Explanation

In the COPY or EMPTY function, the queue manager could not determine which queues are in page set *psid* because the response to a command was not received in time.

System action

The function is terminated.

System programmer response

Check the definitions of the system-command input queue and the system-command reply queue; ensure that they are enabled for MQGET and MQPUT. If the definitions are correct, stop and restart the command server.

CSQU116I
No storage classes found for page set *psid*

Explanation

The page set specified has no storage classes associated with it.

System action

The function is terminated.

System programmer response

Define a storage class for the page set, and rerun the job if required.

CSQU117I
No queues found for page set *psid*

Explanation

The page set specified has no queues associated with it that are eligible for the requested function. For the COPY and EMPTY functions, there are no local queues; for the SCOPY function, there are no local queues with messages.

System action

The function is terminated.

System programmer response

If required, correct the page set specified, and rerun the job.

CSQU120I
Connecting to *conn-id*

Explanation

The utility program is connecting to the named queue manager or queue sharing group.

CSQU121I
Connected to queue manager *qmgr-name*

Explanation

The utility program connected successfully to queue manager *qmgr-name*.

CSQU122I
Executing *function-name*

Explanation

The utility program is executing function *function-name*.

CSQU123I
Processing *ddname* data set, mode FORCE

Explanation

The current function of the utility program is handling data set *ddname* using the FORCE option.

CSQU124I
Processing *ddname* data set

Explanation

The current function of the utility program is handling data set *ddname*.

CSQU125I
n page sets attempted

Explanation

This indicates how many page sets the current function attempted to process.

CSQU126I
n page sets processed successfully

Explanation

This indicates how many page sets were processed successfully by the current function.

CSQU127I
Executing *function* using input from *ddname* data set

Explanation

The utility program is executing function *function* using input from *ddname*.

CSQU128I
Executing *function* outputting to *ddname* data set

Explanation

The utility program is executing function *function*, and is writing the output to *ddname*.

CSQU129I
Copying page set *psid*

Explanation

The utility program is copying page set *psid*.

CSQU130I
Copying queue *q-name*

Explanation

The utility program is copying queue *q-name*.

CSQU131I
n messages copied successfully

Explanation

This indicates how many messages were copied successfully when copying a queue.

CSQU133I
n queues attempted

Explanation

This indicates how many queues the program attempted to copy while copying a page set.

CSQU134I
n queues copied successfully

Explanation

This indicates how many queues were copied successfully while copying a page set.

CSQU135I

Loading queue *sourceq* to *targetq*

Severity
0

Explanation

When executing the LOAD or SLOAD function, identifies the name of the target queue being loaded, and the name of the queue on the input data set from which messages are being copied.

CSQU136I

msg-count messages (*msg-from-msg-to*) have been loaded (total size *text-length*)

Severity
0

Explanation

When executing the LOAD or SLOAD function, this error code indicates that a number of messages have been successfully loaded on to the target queue from the input data set.

- *msg-count* is the number of messages loaded
- *msg-from-msg-to* is the message number range in the messages for the queue on the input data set.
- *text-length* is the total length of the message texts loaded (in MB or KB)

CSQU137I
Skipping queue *q-name*

Explanation

This indicates that queue *q-name* is being bypassed, because of the SKIPQS or FROMQ option used with the LOAD function.

CSQU138I
n queues loaded successfully

Explanation

This indicates how many queues were loaded successfully.

CSQU139I
Emptying page set *psid*

Explanation

The utility program is emptying page set *psid*.

CSQU140I
Emptying queue *q-name*

Explanation

The utility program is emptying queue *q-name*.

CSQU141I
n messages deleted successfully

Explanation

This indicates how many messages were deleted while emptying a queue.

CSQU142I
n queues emptied successfully

Explanation

This indicates how many queues were emptied.

CSQU143I
n function statements attempted

Explanation

This indicates the number of *function* statements attempted by the utility program.

CSQU144I
n function statements executed successfully

Explanation

This indicates the number of *function* statements executed successfully by the utility program.

CSQU145I
function statement failed

Explanation

The utility program experienced an error while executing function *function*.

System action

The utility program terminates.

System programmer response

Check the other messages issued to determine where the error occurred, and what caused it.

CSQU146I

msg-count messages (*msg-from-msg-to*) skipped (total size *text-length*). Reason=*reason-code*

Severity
0

Explanation

When executing the LOAD or SLOAD function, indicates that a number of messages have been ignored from the input data set.

- *msg-count* is the number of messages ignored
- *msg-from-msg-to* is the message number range in the messages for the queue on the input data set.
- *text-length* is the total length of the message texts ignored (in MB or KB)

The reason code indicates why the messages were ignored:

4
messages skipped due to *skipmsg*s parameter in LOAD or SLOAD command

8
messages skipped due to an MQPUT error

12
messages skipped due to an error on MQOPEN

16
messages skipped due to an MQPUT error immediately following a sync point

20
messages skipped due to an error on MQCLOSE

24
messages skipped due to an error when taking a sync point

28
messages skipped due to *MSGCOUNT* limit (in the LOAD or SLOAD command) being reached

CSQU147I
csect-name Utility terminated, return code=*ret-code*

Explanation

The utility has terminated because a severe error or forced syncpoint occurred meaning that no further functions should be run. *ret-code* is the return code from the utility.

System action

The utility ends.

System programmer response

See "IBM MQ for z/OS codes" on page 943 for information about the return code from the utility.

CSQU148I
csect-name Utility completed, return code=*ret-code*

Explanation

The utility completed, all required functions having been attempted. *ret-code* is the return code from the utility.

System action

The utility ends.

System programmer response

Check any functions that failed.

CSQU150I
function completed for data set *ddname1* to data set *ddname2*

Explanation

Processing for data set *ddname1* has completed, with output to *ddname2*.

System action

Processing continues with the next page set.

CSQU151I
No matching CSQSnnnn and CSQTnnnn DD statements. *function* terminated

Explanation

A COPYPAGE or RESETPAGE function was unsuccessful because there were no matching pairs of page set data sets with names CSQS0000 through CSQS0099 and CSQT0000 through CSQT0099.

System action

The function is terminated.

System programmer response

Add DD statements for the required page set data sets, and resubmit the job.

CSQU152I
ddname1 DD statement missing. No action taken for *ddname2* data set

Explanation

Only one of the source-target pair of page set data sets (CSQSnnnn and CSQTnnnn) was specified.

System action

The function continues.

System programmer response

Add DD statements for the required page set data sets, and resubmit the job.

CSQU154E
Target data set *ddname* is smaller than source data set. Function terminated

Explanation

A COPYPAGE or RESETPAGE function could not process a page set data set because the target data set *ddname* was too small.

System action

Processing continues with the next page set.

CSQU155I

Processing queue *queue-name*

Severity
0

Explanation

When executing the ANALYZE function, indicates the start of processing queue *queue-name* from the input data set.

CSQU156E
GET failed for *ddname* data set. VSAM return code=*rc* reason code=*code*

Explanation

The utility program received a VSAM GET error for the page set it was attempting to process (pointed to by *ddname*).

System action

Processing for the page set is terminated, and the function continues with the next page set.

System programmer response

See the *z/OS DFSMS Macro Instructions for Data Sets* manual for information about the return and reason codes from VSAM. If necessary, resubmit the job.

CSQU157I
Processing data set *ddname1* to *ddname2*

Explanation

The current function is handling data set *ddname1*, with output to *ddname2*.

CSQU158E
Target data set *ddname2* is not newly formatted

Explanation

The COPYPAGE and RESETPAGE functions can only be used with a newly formatted target page set.

System action

Processing continues with the next page set.

System programmer response

Specify a valid target page set, and resubmit the job.

CSQU159E
Source data set *ddname1* is not a page set

Explanation

CSQUTIL COPYPAGE or RESETPAGE functions were unable to recognize the data set as an IBM MQ Page set. This could be due to an invalid data set, or a back level version of IBM MQ libraries being used.

System action

Processing continues with the next page set.

System programmer response

Check the data set is a valid IBM MQ page set.

Check the IBM MQ libraries being used are the same as the libraries used by the queue manager.

CSQU160E
Data set *ddname* is not suitable for use with the function

Explanation

The function should only be used with page sets for a queue manager that terminated normally.

System action

Processing continues with the next page set.

System programmer response

Specify a valid page set, and resubmit the job.

CSQU161I
ddname contains *pp* pages and was formatted as page set *nn*

Explanation

This is part of the response to the PAGEINFO function for data set *ddname*.

It shows the size of the page set, and the page set number that was assumed when it was formatted. The number is derived from the DD name used when formatting, which was CSQP00*nn*.

CSQU162I
ddname is used as page set *psid* for queue manager *qmgr-name*

Explanation

This is part of the response to the PAGEINFO function for data set *ddname*.

The page set has been used by the queue manager shown. The page set number is not necessarily the same as that with which it was formatted, as shown in message CSQU161I.

CSQU163I
ddname has page set recovery RBA = *rba*

Explanation

This is part of the response to the PAGEINFO function for data set *ddname*.

CSQU164I
ddname System recovery RBA for all page sets successfully processed = *rba*

Explanation

This is part of the response to the PAGEINFO function. Note that this RBA relates only to those page sets processed; it does not relate to the whole queue manager unless all the page sets for the queue manager are included.

CSQU165I
Processing *ddname* data set, TYPE(*type*)

Explanation

This current function of the utility program is handling data set *ddname* with the options shown.

CSQU166I
Processing *ddname* data set, TYPE(*type*), mode FORCE

Explanation

This current function of the utility program is handling data set *ddname* with the options shown.

CSQU167I
ddname has never been initialized by a queue manager

Explanation

This is part of the response to the PAGEINFO function for data set *ddname*.

CSQU168E
Requested page sets are for more than one queue manager

Explanation

The page sets for which information was requested are associated with more than one queue manager. No system recovery RBA can therefore be determined.

System action

Processing continues.

System programmer response

Specify a set of page sets for a single queue manager, and resubmit the job.

CSQU169E

MQPUT of message *msg-no* failed. MQCC=*mqcc* MQRC=*mqrc (mqrc-text)*

Severity

8

Explanation

When executing the LOAD or SLOAD function, an MQPUT failed for message number *msg-no* in the queue currently being processed on the input data. The *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form) indicate the reason for failure.

System action

Processing for the command is terminated.

System programmer response

Using the MQ completion code and reason code in the message, determine the cause of error and correct the problem. Then rerun the LOAD or SLOAD, starting with the queue being processed at the time of the error. If any messages had been successfully loaded from the input queue before the failure, use the *SKIPMSGS* parameter on the LOAD or SLOAD command to bypass those messages.

CSQU170I

msg-count messages (*msg-from*-*msg-to*) found (total size *text-length*)

Severity

0

Explanation

When executing the ANALYZE function, this message is displayed for the queue being processed from the input data set. The number of messages and the total length of the message text are shown.

CSQU171E

Queue *queue-name* was not found in the input data set

Severity

8

Explanation

The LOAD or SLOAD function being executed specified a source queue name of *queue-name* which was not found on the input data set.

System action

Processing for the command is terminated.

System programmer response

Specify the correct input file, correct queue name in the command, and try again.

CSQU172I

Processing *function-name* for data set *ddname*, *current-page* of *total-pages* pages processed, *percentage*% complete

Explanation

If a CSQUTIL function to process a page set is long-running, this message is issued periodically to indicate how many pages have been processed so far.

CSQU179E

The transmission queue cannot be switched because the channel initiator is not active

Severity

8

Explanation

The utility program is unable to initiate a switch of transmission queue for the channel identified in the preceding CSQU183I message because the channel initiator is not active.

System action

Processing continues, however, the transmission queue for the affected channel is not switched.

System programmer response

Start the channel initiator, then either restart the channel or rerun the command to initiate the switch of transmission queue.

CSQU182E

An error occurred obtaining the list of channels, reason *mqrc (mqrc-text)*

Severity
8

Explanation

The utility program was unable to identify the list of channels to process.

System action

Processing for the command is terminated.

System programmer response

Use the reason code to identify and resolve the error, then rerun the command if required.

See “API completion and reason codes” on page 1151 for information about *mqrc* (*mqrc-text* provides the MQRC in textual form).

CSQU183I

Channel *channel-name*

Severity
0

Explanation

The utility program is processing the requested function for the channel named *channel-name*.

System action

Processing continues.

System programmer response

Check subsequent messages to determine whether the requested function was processed successfully.

CSQU184I

n channels processed

Severity
0

Explanation

This message identifies the number of channels that were processed by the requested function.

System action

Processing continues.

System programmer response

None required.

CSQU185I

A switch of transmission queue is not required for this channel

Severity
0

Explanation

This message identifies that a switch of transmission queue is not required for the channel identified in the preceding CSQU183I message. This indicates the channel uses the currently configured transmission queue.

System action

Processing continues.

System programmer response

None required.

CSQU186I

A switch of transmission queue is pending for this channel

Severity
0

Explanation

This message identifies that a switch of transmission queue is pending for the channel identified in the preceding CSQU183I message. This indicates the configured transmission queue for the channel has been changed, either by updating the **DEFCLXQ** queue manager attribute, or by altering the value of the **CLCHNAME** attribute of a transmission queue, since the channel last started.

The switch of transmission queue will occur the next time the channel starts or if the switch is initiated using the CSQUTIL function **SWITCH CHANNEL**.

A switch operation is also reported as pending if the operation was previously initiated, but the queue manager was stopped while messages were being moved from the old transmission queue to the new transmission queue. To resume the switch operation either start the channel or use CSQUTIL to initiate the switch.

System action

Processing continues.

System programmer response

None required.

CSQU187I

A switch of transmission queue is in progress for this channel

Severity
0

Explanation

This message identifies that a switch of transmission queue isin progress for the channel identified in the preceding CSQU183I message.

System action

Processing continues.

System programmer response

Use console messages issued by the queue manager to determine the status of the switch operation, if required.

CSQU188I

From transmission queue *xmit-qname*

Severity
0

Explanation

This message is issued with other messages, such as CSQU186I, CSQU187I, and CSQU195I. It identifies the name of the transmission queue a channel is switching, or will switch, from.

System action

Processing continues.

System programmer response

None required.

CSQU189I

To transmission queue *xmit-qname*

Severity
0

Explanation

This message is issued with other messages, such as CSQU186I, CSQU187I, and CSQU195I. It identifies the name of the transmission queue a channel is switching, or will switch, to.

System action

Processing continues.

System programmer response

None required.

CSQU190I

There are *num-msgs* messages queued for this channel on *xmitq-name*

Severity
0

Explanation

This message is issued with CSQU186I and identifies there are currently *num-msgs* messages queued for the channel on the transmission queue *xmitq-name*, that need to be moved when the transmission queue is switched.

System action

Processing continues.

System programmer response

None required.

CSQU191E

Unable to access transmission queue *xmitq-name*, reason *mqrc (mqrc-text)*

Severity
8

Explanation

This message is issued with CSQU186I if the transmission queue, *xmitq-name*, that is currently used by the channel, cannot be accessed for the reason *mqrc* (*mqrc-text* provides the MQRC in textual form). This transmission queue must be accessible to move messages for the channel to the new transmission queue.

System action

Processing for the command is terminated.

System programmer response

Use the reason code to identify and resolve the error, then rerun the command if required.

See “API completion and reason codes” on page 1151 for information about *mqrc* (*mqrc-text* provides the MQRC in textual form).

Alternatively, use the CSQUTIL function **SWITCH CHANNEL** with the **MOVEMSGS(NO)** option to switch transmission queue without moving messages. If this option is selected it is the responsibility of the system programmer to resolve any messages for the channel on the transmission queue, *xmitq-name*, after the switch has completed.

CSQU192E

The status of this channel is unavailable, reason *mqrc (mqrc-text)*

Severity
8

Explanation

The utility program was unable to identify the current status of the channel identified in the preceding CSQU183I message to determine if a switch of transmission queue is pending or in progress.

System action

Processing continues, but the transmission queue for the affected channel is not switched if this was requested.

System programmer response

Use the reason code to identify and resolve the error, then rerun the command if required.

See “API completion and reason codes” on page 1151 for information about *mqrc* (*mqrc-text* provides the MQRC in textual form).

CSQU193E

The transmission queue cannot be switched because the channel is active

Severity
8

Explanation

The utility program was unable to initiate a switch of transmission queue for the channel identified in the preceding CSQU183I message because the channel status was neither **STOPPED** nor **INACTIVE**.

If the cluster sender channel is showing STOPPED status, but message CSQU193E is still reporting, the channel cannot stop immediately. Message reallocation is taking place while a request to STOP CHANNEL *channel-name* is made.

The channel continues to reallocate messages, and stops once this process is complete. This process can take some length of time if there are a large number of messages on the queue assigned to this channel.

You should wait for sufficient time to ensure that message reallocation completes, then switch the transmission queue.

System action

Processing continues, but the transmission queue for the affected channel is not switched.

System programmer response

Stop the channel, then either restart the channel or rerun the command to initiate the switch of transmission queue.

CSQU194E

The switch of transmission queue failed, reason *mqrc (mqrc-text)*

Severity
8

Explanation

The utility program was unable to switch the transmission queue for the channel identified in the preceding CSQU183I message.

System action

Processing continues, but the transmission queue for the affected channel is not switched.

System programmer response

Use the reason code to identify and resolve the error, then rerun the command if required.

See "API completion and reason codes" on page 1151 for information about *mqrc* (*mqrc-text* provides the MQRC in textual form).

CSQU195I

Switching the transmission queue for this channel

Severity
0

Explanation

A switch of transmission queue has been initiated for the channel identified in the preceding CSQU183I message.

System action

Processing continues.

System programmer response

Use subsequent messages to determine if the switch of transmission queue completes successfully

CSQU196I

Moving messages for this channel - *num-msgs* messages moved

Severity
0

Explanation

A switch of transmission queue requires that messages for a channel are moved from the old transmission queue to the new transmission queue. This message is periodically issued during a switch of transmission queue to report the progress of this operation for the channel identified in the preceding CSQU183I message.

System action

Processing continues.

System programmer response

If this message is repeatedly issued it might indicate the old transmission queue cannot be drained of messages for the channel, which means the switching process can not complete. Applications continue to put messages to the old transmission queue during the switching process to preserve ordering.

If the switching process cannot complete, this might indicate that messages are being put to the old transmission queue faster than they can be moved by the switching process, or uncommitted messages remain on the old transmission queue for the channel.

Use console messages issued by the queue manager, such as CSQM554I, and commands such as **DISPLAY QSTATUS**, to determine why the switch operation is unable to complete.

CSQU197I

Moving of messages complete - *num-msgs* messages moved

Severity
0

Explanation

A switch of transmission queue requires that messages for a channel are moved from the old transmission queue to the new transmission queue. This message indicates the process of moving messages for the channel identified in the preceding CSQU183I message has completed. The number of messages that were moved to the new transmission queue is identified by *num-msgs*.

System action

Processing continues.

System programmer response

None required.

CSQU198I

The transmission queue has been switched successfully

Severity
0

Explanation

A switch of transmission queue for the channel identified in the preceding CSQU183I message has completed successfully.

System action

Processing continues.

System programmer response

None required.

CSQU199E

Function requires command level *required-cmdlevel*, the queue manager's command level is *qmgr-cmdlevel*

Severity
8

Explanation

The utility program was unable to perform the requested function, identified by *function*, because this is not supported by the queue manager to which it is connected. To perform the requested function the queue manager's command level must be *required-cmdlevel* or greater, but the queue manager's command level is *qmgr-cmdlevel*.

System action

Processing for the command is terminated.

System programmer response

Ensure the utility program connected to the required queue manager. If this was correct, the queue manager must be upgraded before the requested function can be used.

CSQU200I
csect-name Dead-letter Queue Handler Utility - *date time*

Explanation

This is part of the header to the report issued by the utility program.

CSQU201I
Processing queue *q-name*

Explanation

The dead-letter queue handler has parsed the rules table without detecting any errors and is about to start processing the queue identified in the message.

CSQU202I
Dead-letter queue handler ending. Successful actions: *n1* retries, *n2* forwards, *n3* discards

Explanation

The dead-letter queue handler is ending because there are no more messages on the dead-letter queue, or because the queue manager is shutting down, or because the dead-letter queue handler detected an error. The message indicates how many dead-letter queue messages were successfully handled.

System action

The utility terminates.

System programmer response

If the utility ended because of an error, investigate the problem reported in the preceding messages.

CSQU203I
n messages remain on the dead-letter queue

Explanation

The message indicates how many messages are left on the dead-letter queue when the dead-letter queue handler ends.

CSQU210I
Message does not have a valid MQDLH

Explanation

The dead-letter queue handler retrieved a message from the dead-letter queue, but the message was not prefixed by a valid dead-letter queue header (MQDLH). This typically occurs because an application is writing directly to the dead-letter queue but is not prefixing messages with a valid MQDLH.

System action

The message is left on the dead-letter queue and the dead-letter queue handler continues to process the dead-letter queue.

This message is issued only once the first time such a message is encountered.

System programmer response

Remove all the invalid messages from the dead-letter queue. Do not write messages to the dead-letter queue unless they are prefixed by a valid MQDLH.

CSQU211I
Unable to put message, line *n* MQRC=*mqrc (mqrc-text)*

Explanation

The dead-letter queue handler tried to redirect a message to another queue as requested, but the MQPUT call was unsuccessful.

System action

The retry count for the message is incremented; processing continues.

System programmer response

Refer to “API completion and reason codes” on page 1151 for information about *mqrc* (*mqrc-text* provides the MQRC in textual form). The line number *n* of the rules table used to determine the action for the message will help identify the queue to which the message was being put.

CSQU212I
Unable to inquire dead-letter queue, MQCC=*mqcc* MQRC=*mqrc (mqrc-text)*

Explanation

An MQINQ call for the dead-letter queue was unsuccessful.

System action

Processing continues.

System programmer response

Refer to “API completion and reason codes” on page 1151 for information about *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form).

CSQU213I
Unable to convert message, MQCC=*mqcc* MQRC=*mqrc (mqrc-text)*

Explanation

An MQGET call encountered a data conversion problem.

System action

The message is rolled back and remains on the queue. Processing of the remaining messages on the queue continues. Use an alternative means to remove this message from the dead-letter queue.

System programmer response

Refer to “API completion and reason codes” on page 1151 for information about *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form).

CSQU220E
Unable to connect to queue manager qmgr-name, MQCC=*mqcc* MQRC=*mqrc (mqrc-text)*

Explanation

The dead-letter queue handler could not connect to the requested queue manager.

System action

The utility is terminated.

System programmer response

Refer to “API completion and reason codes” on page 1151 for information about *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form).

CSQU221E
Unable to open queue manager, MQCC=*mqcc* MQRC=*mqrc (mqrc-text)*

Explanation

An MQOPEN call for the queue manager was unsuccessful.

System action

The utility is terminated.

System programmer response

Refer to “API completion and reason codes” on page 1151 for information about *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form).

CSQU222E
Unable to inquire queue manager, MQCC=*mqcc* MQRC=*mqrc (mqrc-text)*

Explanation

An MQINQ call for the queue manager was unsuccessful.

System action

The utility is terminated.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form).

CSQU223E
Unable to close queue manager, MQCC=*mqcc* MQRC=*mqrc (mqrc-text)*

Explanation

An MQCLOSE call for the queue manager was unsuccessful.

System action

The utility is terminated.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form).

CSQU224E
Unable to browse dead-letter queue *q-name*, MQCC=*mqcc* MQRC=*mqrc (mqrc-text)*

Explanation

An MQOPEN call for browsing the dead-letter queue was unsuccessful. This is typically because of one of the following reasons:

- Another process has opened the queue for exclusive access.
- An invalid queue name was specified.
- The alias name for one of the following modules has been lost:
  - CSQBSRV
  - CSQAPEPL
  - CSQBCRMH
  - CSQBAPPL

System action

The utility is terminated.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form).

CSQU225E
Unable to close dead-letter queue, MQCC=*mqcc* MQRC=*mqrc (mqrc-text)*

Explanation

An MQCLOSE call for the dead-letter queue was unsuccessful.

System action

The utility is terminated.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form).

CSQU226E
Line *n*: *keyword*(*value*) invalid or outside permitted range

Explanation

The value supplied for the specified keyword in line *n* of the rules table was outside the valid range of values or otherwise invalid.

System action

The utility is terminated.

System programmer response

Correct the rules table and restart the dead-letter queue handler.

CSQU227E
Unable to get message from dead-letter queue, MQCC=*mqcc* MQRC=*mqrc (mqrc-text)*

Explanation

An MQGET call for the dead-letter queue was unsuccessful.

System action

The utility is terminated.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form).

CSQU228E
Unable to commit or backout dead-letter queue action, MQCC=*mqcc* MQRC=*mqrc (mqrc-text)*

Explanation

An MQCMIT or MQBACK call for the dead-letter queue was unsuccessful.

System action

The utility is terminated.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form).

CSQU229E
Rules table is invalid or missing

Explanation

The rules table contained no valid message templates or was not supplied at all.

System action

The utility is terminated.

System programmer response

Correct the rules table as indicated in the preceding messages and restart the dead-letter queue handler.

CSQU230E
Unable to obtain storage

Explanation

The dead-letter queue handler was unable to obtain storage.

This problem would typically arise as a result of some wider problem. For example, if there is a persistent problem that is causing messages to be written to the dead-letter queue and the same

problem (for example, queue full) is preventing the dead-letter queue handler from taking the requested action with the message, ever-increasing amounts of storage would be required.

System action

The utility is terminated.

System programmer response

Increase the storage available to the utility. Investigate whether some wider problem exists, and if the dead-letter queue contains a large number of messages.

CSQU231E
Line *n*: parameter *keyword* exceeds maximum length

Explanation

The value for the specified parameter in line *n* of the rules table is too long.

System action

The utility is terminated.

System programmer response

Correct the rules table and restart the dead-letter queue handler.

CSQU232E
Line *n*: parameter *keyword* is duplicated

Explanation

Two or more parameters of the same type were supplied in line *n* of the rules table.

System action

The utility is terminated.

System programmer response

Correct the rules table and restart the dead-letter queue handler.

CSQU233E
Line *n*: syntax error

Explanation

There is a syntax error in line *n* of the rules table.

System action

The utility is terminated.

System programmer response

Correct the rules table and restart the dead-letter queue handler.

CSQU234E
Unable to release storage

Explanation

The dead-letter queue handler was unable to release storage.

System action

The utility is terminated.

System programmer response

Investigate the problem reported in the preceding messages.

CSQU235E
Line *n*: *keyword* value invalid or outside permitted range

Explanation

The value supplied for the specified parameter in line *n* of the rules table was outside the valid range of values or otherwise invalid.

System action

The utility is terminated.

System programmer response

Correct the rules table and restart the dead-letter queue handler.

CSQU236E
n error(s) in rules table

Explanation

Errors were detected in the rules table.

System action

The utility is terminated.

System programmer response

Correct the rules table as indicated in the preceding messages and restart the dead-letter queue handler.

CSQU237E
Line *n*: invalid keyword combination

Explanation

There is an invalid combination of parameters in line *n* of the rules table. For example: no ACTION specified, ACTION(FWD) specified without FWDQ, HEADER specified without ACTION(FWD).

System action

The utility is terminated.

System programmer response

Correct the rules table and restart the dead-letter queue handler.

CSQU249E
Unable to disconnect from queue manager, MQCC=*mqcc* MQRC=*mqrc (mqrc-text)*

Explanation

An MQDISC call for the queue manager was unsuccessful.

System action

The utility is terminated.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form).

CSQU300I
Incorrect parameters provided. Usage information follows:

CSQUDSPM (-m QMgrName) (-o status | -s) (-o all) (-su) (-a)

-m QMgrName: Display information for this queue manager only

-o status : Display operational status of the queue manager, or queue managers

-s : Display operational status of the queue manager, or queue managers

-o all : Display all details of the queue manager, or queue managers

-su : Do not show information for queue managers that have an unknown version

-a : Display information for running queue managers only

Explanation

Incorrect parameters were passed to CSUDSPM.

System action

The utility stops processing.

System programmer response

Correct the parameters and resubmit the utility.

CSQU307I
No queue manager with the specified name exists

Explanation

The CSQUDSPM utility was run, specifying a queue manager name that could not be located.

System action

The utility stops processing.

System programmer response

Either correct the queue manager name, or run the utility without specifying a particular queue manager.

CSQU500I
csect-name Queue Sharing Group Utility - *date time*

Explanation

This is part of the header to the report issued by the utility program.

CSQU501I
function function requested

Explanation

This identifies the utility function requested.

CSQU502I
Queue manager=*qmgr-name*

Explanation

This identifies the queue manager name for which the function is requested.

CSQU503I
QSG=*qsg-name*, Db2 DSG=*dsg-name*, Db2 ssid=*db2-name*

Explanation

This identifies the queue sharing group, Db2 data-sharing group, and Db2 subsystem names for which the function is requested.

CSQU504E
Unable to LOAD *module-name*, reason=*ssssrrrr*

Explanation

The utility was unable to load a required module. *ssss* is the completion code and *rrrr* is the reason code (both in hexadecimal) from the z/OS LOAD service.

System action

The utility terminates.

System programmer response

Check the console for messages indicating why the module was not loaded. See the *z/OS MVS Programming: Assembler Services Reference* Volume 2 manual for information about the codes from the LOAD request.

Ensure that the module is in the required library, and that it is referenced correctly. The utility attempts to load this module from the library data sets under the STEPLIB DD statement.

CSQU505E
No EXEC PARM parameters

Explanation

No parameters for the utility were specified in EXEC PARM field.

System action

The utility program is terminated.

System programmer response

Specify the required parameters and rerun the job.

CSQU506E
Invalid EXEC PARM function parameter

Explanation

The function requested for the utility, as the first parameter in EXEC PARM field, was invalid.

System action

The utility program is terminated.

System programmer response

Correct the parameter and rerun the job.

CSQU507E
Wrong number of EXEC PARM parameters for function

Explanation

The number of parameters for the utility specified in EXEC PARM field was incorrect for the function requested.

System action

The utility program is terminated.

System programmer response

Correct the parameters and rerun the job.

CSQU508E
Invalid EXEC PARM parameter *n*

Explanation

The *n*th parameter for the utility specified in EXEC PARM field was invalid for the function requested, or omitted but required by the function requested.

System action

The utility program is terminated.

System programmer response

Correct the parameter and rerun the job.

CSQU509E
Too many EXEC PARM parameters

Explanation

The number of parameters for the utility specified in EXEC PARM field was too many for the function requested.

System action

The utility program is terminated.

System programmer response

Correct the parameters and rerun the job.

CSQU510I
SQL error information

Explanation

An SQL error has occurred. Diagnostic information follows in message CSQU511I.

System action

See the preceding utility error message.

System programmer response

Look at the information in message CSQU511I to determine the reason for the SQL error.

CSQU511I
DSNT408I SQLCODE = `-sql-code`, explanation

Explanation

This message provides additional diagnostic information direct from Db2. It is followed by further CSQU511I messages, similar to the following example block:

```
CSQU511I   DSNT408I SQLCODE = -805, ERROR:  DBRM OR PACKAGE NAME DSNV11P1..CSQ5B9-
CSQU511I            00.1A47B13F08B31B99 NOT FOUND IN PLAN CSQ5B900. REASON 03
CSQU511I   DSNT418I SQLSTATE   = 51002 SQLSTATE RETURN CODE
CSQU511I   DSNT415I SQLERRP    = DSNXEPM SQL PROCEDURE DETECTING ERROR
CSQU511I   DSNT416I SQLERRD    = -251  0  0  -1  0  0 SQL DIAGNOSTIC INFORMATION
CSQU511I   DSNT416I SQLERRD    = X'FFFFFF05'  X'00000000'  X'00000000'
CSQU511I            X'FFFFFFFF'  X'00000000'  X'00000000' SQL DIAGNOSTIC
CSQU511I            INFORMATION
```

System action

The utility program is terminated.

System programmer response

Use the diagnostic information to determine the reason for the SQL error and correct the problem.

CSQU512E
Utility terminated, Db2 tables in use

Explanation

The queue sharing group utility cannot run because the Db2 tables it uses are reserved by another job. The most likely reason is that another instance of the utility is running, or that a queue manager in the queue sharing group is in the process of starting.

System action

The utility program is terminated.

System programmer response

Rerun the job later.

CSQU513E
Utility terminated, not APF authorized

Explanation

The queue sharing group utility is not APF authorized.

System action

The utility program is terminated.

System programmer response

Ensure that the library data sets under the STEPLIB DD statement comply with the rules for APF authorization, and rerun the job.

CSQU514E
RRSAF function *call-name* failed, RC=*rc*

Explanation

The RRS function specified by *call-name* returned an unexpected reason code specified by *rc*.

System action

The utility program is terminated.

System programmer response

See the *Db2for z/OS* information for an explanation of the RRSAF reason code.

Take corrective action if necessary and resubmit the job.

CSQU515E
Unable to access Db2 tables, RC=*rc* reason=*reason*

Explanation

The call to CSQ5ARO2 module failed with a return code specified by *rc* and reason code specified by *reason*.

System action

The utility program is terminated.

System programmer response

Resubmit the job. If the problem persists, note the error codes in the message and contact IBM Support.

CSQU517I
XCF group *xcf-name* already defined

Explanation

Informational message indicating that the XCF group name specified by *xcf-name* already exists.

CSQU518E
XCF IXCQUERY member error, RC=*rc* reason=*reason*

Explanation

An unexpected return code specified by *rc* with reason code specified by *reason* was returned from an IXCQUERY request.

System action

The utility program is terminated.

System programmer response

See the *z/OS MVS Sysplex Services Reference* manual for an explanation of the IXCQUERY return and reason codes.

Take corrective action if necessary and resubmit the job.

CSQU520I
Summary information for XCF group *xcf-name*

Explanation

Informational message indicating that summary data for the XCF group specified by *xcf-name* follows.

CSQU521I
Group contains *n* members:

Explanation

Informational message indicating that the group specified by message CSQU517I contains *n* members.

CSQU522I
Member=*xcf-name*, state=*sss*, system=*sys-name*

Explanation

Informational message indicating that the XCF group member specified by *xcf-name* has a state of *sss* and last executed on system *sys-name*.

CSQU523I
User data=*xxx*

Explanation

Informational message containing the 32 bytes of XCF user data to accompany informational message CSQU522I.

CSQU524I
QMGR number=nn

Explanation

Informational message containing the number of the QMGR in the queue sharing group to accompany informational message CSQU522I. The QMGR number is stored in the Db2 tables, the XCF group member and the connections to the CF structures. It is generated when a QMGR is added to a queue sharing group using CSQ5PQSG.

CSQU525E
Db2 *db2-name* is not a member of data-sharing group *dsg-name*

Explanation

There was an inconsistency between the Db2 ssid and data-sharing group name provided in the EXEC PARM field. Db2 ssid specified by *db2-name* is not a member of the Db2 data-sharing group specified by *dsg-name*.

System action

The utility program is terminated.

System programmer response

Ensure that the Db2 ssid specified is a member of the Db2 data-sharing group specified.

Issue the following command to find the group member name: DISPLAY XCF,GROUP

CSQU526I
Connected to Db2 *db2-name*

Explanation

The utility program connected successfully to Db2 subsystem *db2-name*.

CSQU527E
No eligible Db2 currently active

Explanation

If a Db2 ssid was specified in the EXEC PARM field this indicates that the Db2 subsystem is not currently active on the z/OS system on which the utility job executed.

If a Db2 data-sharing group name was specified in the EXEC PARM field then no eligible Db2 subsystem was active on the z/OS system on which the utility job executed.

System action

The utility program is terminated.

System programmer response

If a Db2 ssid was specified in the EXEC PARM field then ensure that it is active on the z/OS system on which the utility job will execute.

If a Db2 data-sharing group name was specified in the EXEC PARM field then ensure that at least one eligible Db2 subsystem is active on the z/OS system on which the utility job will execute.

CSQU528I
Disconnected from Db2 *db2-name*

Explanation

The utility program disconnected successfully from Db2 subsystem *db2-name*.

CSQU529E
QSG *qsg-name* entry cannot be removed, *n* members are still defined

Explanation

A request to remove the queue sharing group name in *qsg-name* failed because *n* members are still defined to it.

System action

The utility program is terminated.

System programmer response

All members of the queue sharing group must be removed from it before the queue sharing group itself can be deleted. Use the preceding CSQU522I message to identify which queue sharing group members are still defined to the queue sharing group.

Note: Members in a state of ACTIVE or FAILED cannot be removed from a queue sharing group.

CSQU530E
QMGR *qmgr-name* entry cannot be removed from queue sharing group *qsg-name*, status is *sss*

Explanation

The queue manager named by *qmgr-name* cannot be removed from the queue sharing group named by *qsg-name* because it is in an incorrect XCF member state as specified by *sss*.

System action

The utility program is terminated.

System programmer response

To remove a queue manager from the queue sharing group it must have XCF member state CREATED or QUIESCED.

If the XCF member state is ACTIVE then stop the queue manager with a STOP QMGR command and resubmit the job.

If the XCF member state is FAILED then start the queue manager and stop it normally using the STOP QMGR command and resubmit the job.

CSQU531E
QSG *qsg-name* entry cannot be removed, not found in Db2 table *table-name*

Explanation

An attempt to remove the queue sharing group *qsg-name* failed because no entry for it was found in the Db2 table *table-name*.

System action

The utility program is terminated.

System programmer response

Ensure that the queue sharing group *qsg-name* was originally defined in the table *table-name*.

Check that the utility job connected to the correct Db2 data-sharing group. If necessary resubmit the job.

CSQU532E
QSG *qsg-name* entry cannot be deleted, Db2 entries still exist for it

Explanation

An attempt to remove the queue sharing group *qsg-name* was returned a Db2 constraint failure because queue manager entries still exist in the CSQ.ADMIN_B_QMGR table.

System action

The utility program is terminated.

System programmer response

Examine the CSQ.ADMIN_B_QMGR table to determine which queue managers are still defined to the queue sharing group *qsg-name*.

Use the REMOVE QMGR function of the CSQ5PQSG utility to remove the entries and then resubmit the job.

CSQU533E
SQL error. Db2 table=*table-name*, code=*sqlcode*, state=*sss*, data=*sqlerrcd*

Explanation

An unexpected SQL error was returned from Db2. An operation on the table named by *table-name* was returned an SQLCODE specified by *sqlcode* with STATE specified by *sss* and SQLERRCD values specified by *sqlerrcd*.

System action

The utility program is terminated.

System programmer response

See the *Db2for z/OS* information for an explanation of the SQL codes.

Resubmit the job if required.

CSQU534E
SQL services error, Db2 table=*table-name* RC=rc

Explanation

An SQL error occurred during an operation on the table specified by *table-name*, as reported in the preceding CSQU533E message. A return code of *rc* was returned from the internal service routine.

System action

The utility program is terminated.

System programmer response

See message CSQU533E.

CSQU535I
QSG *qsg-name* entry successfully removed from Db2 table *table-name*

Explanation

Informational message indicating that the queue sharing group named by *qsg-name* was successfully removed.

CSQU536E
Unable to add queue sharing group *qsg-name* entry, entry already exists in Db2 table *table-name*

Explanation

An attempt to add the queue sharing group *qsg-name* failed because an entry already exists in the Db2 table *table-name*.

System action

The utility program is terminated.

CSQU537I
csect-name queue sharing group *qsg-name* entry successfully added to Db2 table *table-name*

Explanation

The request to add the queue sharing group *qsg-name* to the Db2 table *table-name* completed successfully.

CSQU538E
Member record found for QMGR *qmgr-name* XCF group *xcf-name*

Explanation

Informational message indicating that a member record for the queue manager named in *qmgr-name* already exists in the XCF group named by *xcf-name*.

CSQU539E
No QMGR *qmgr-name* entry found in queue sharing group *qsg-name*

Explanation

An attempt to remove the queue manager named by *qmgr-name* from the queue sharing group named by *qsg-name* failed because no entry was found in the Db2 tables.

System action

The utility program is terminated.

CSQU540E
Unable to remove QMGR *qmgr-name* - not terminated normally, or needed for recovery

Explanation

The queue manager named by *qmgr-name* cannot be removed from the queue sharing group because it is currently active, or because it terminated abnormally during its last execution, or because it is needed for backup and recovery purposes.

System action

The utility program is terminated.

System programmer response

If the queue manager is active then stop the queue manager with a STOP QMGR command and resubmit the job.

If the queue manager terminated abnormally during its last execution then start the queue manager and stop it normally using the STOP QMGR command and resubmit the job.

If neither of these cases applies, or if it still cannot be removed, it must be needed for backup and recovery purposes. See Managing queue sharing groups for information about removing such a queue manager from a queue sharing group.

CSQU541E
QSG array manipulation error, RC=`rc`

Explanation

An internal error occurred during manipulation of the queue sharing group array data.

An internal routine returned a completion code specified by *rc*.

System action

The utility program is terminated.

System programmer response

Resubmit the job. If the problem persists, note the error codes in the message and contact IBM Support.

CSQU542E
Update unsuccessful for queue sharing group *qsg-name*, RC=*rc*

Explanation

An attempt to update the Db2 row for the queue sharing group named by *qsg-name* failed with return code *rc*.

rc shows the type of failure:

00F5000C
Queue sharing group row no longer exists

00F50010
Internal error

00F50018
Referential constraint failure

00F50028
Internal error

System action

The utility program is terminated.

System programmer response

Resubmit the job. If the problem persists contact IBM Support.

CSQU543E
Delete unsuccessful for QMGR *qmgr-name*, RC=*rc*

Explanation

The attempt to delete the queue manager *qmgr-name* failed with return code *rc*.

rc shows the type of failure: 00F5000C, queue manager row no longer exists.

System action

Processing continues.

System programmer response

This might be an indication that the request was made against the wrong Db2 data-sharing group or that a previous attempt terminated prematurely. For the former, the utility should be executed against the correct Db2 data-sharing group. For the latter, no further action need be taken.

CSQU544E
IXCDELET request for QMGR *qmgr-name* unsuccessful, RC=*rc* reason=*reason*

Explanation

During an attempt to delete queue manager *qmgr-name*, an IXCDELET request was returned an IXC return code of *rc* and reason code of *reason*.

System action

The utility program is terminated.

System programmer response

See the *z/OS MVS Sysplex Services Reference* manual for an explanation of the IXCDELET return and reason codes.

Take corrective action if necessary and resubmit the job.

CSQU545E
IXCCREAT request for QMGR *qmgr-name* unsuccessful, RC=*rc* reason=*reason*

Explanation

During an attempt to add queue manager *qmgr-name*, an IXCCREAT request was returned an IXC return code of *rc* and reason code of *reason*.

System action

The utility program is terminated.

System programmer response

See the *z/OS MVS Sysplex Services Reference* manual for an explanation of the IXCCREAT return and reason codes.

Take corrective action if necessary and resubmit the job.

CSQU546E
Unable to add QMGR *qmgr-name* entry, already exists in Db2 table *table-name*

Explanation

The attempt to add an entry for queue manager *qmgr-name* to the Db2 table *table-name* failed because a row already exists for the queue manager.

System action

The utility program is terminated.

System programmer response

Examine the Db2 table specified by *table-name* and determine whether the entry for the queue manager specified by *qmgr-name* is for the correct queue sharing group. If it is, no further action is required.

CSQU547E
Unable to add QMGR *qmgr-name* entry, no queue sharing group *qsg-name* entry exists in Db2 table *table-name*

Explanation

The attempt to add queue manager *qmgr-name* failed because there is no queue sharing group entry for the queue sharing group *qsg-name* in the Db2 table *table-name*.

System action

The utility program is terminated.

System programmer response

To add a queue manager to a queue sharing group the Db2 CSQ.ADMIN_B_QSG table must contain a queue sharing group record for the queue sharing group named by *qsg-name*.

Examine the Db2 tables and if necessary run the CSQ5PQSG utility ADD QSG function prior to resubmitting this job.

CSQU548E
QMGR *qmgr-name* cannot be added to queue sharing group *qsg-name*, no unassigned QMGR number

Explanation

The attempt to add queue manager *qmgr-name* to the queue sharing group *qsg-name* failed because all queue manager numbers are in use.

System action

The utility program is terminated.

System programmer response

A maximum of 32 queue managers can be defined to a queue sharing group at any one time. If the queue sharing group named by *qsg-name* already contains 32 queue managers then the only course of action is to create a new queue sharing group or remove an existing queue manager.

CSQU549I
QMGR *qmgr-name* entry successfully added to QSG *qsg-name*

Explanation

The request to add queue manager *qmgr-name* to the queue sharing group *qsg-name* completed successfully.

CSQU550I
QMGR *qmgr-name* entry successfully removed from QSG *qsg-name*

Explanation

The request to remove queue manager *qmgr-name* from the queue sharing group *qsg-name* completed successfully.

CSQU551I
QSG *qsg-name* entry successfully added

Explanation

The request to add queue sharing group *qsg-name* completed successfully.

CSQU552I
QSG *qsg-name* entry successfully removed

Explanation

The request to remove queue sharing group *qsg-name* completed successfully.

CSQU553E
QMGR *qmgr-name* exists in Db2 table *table-name* as a member of a different queue sharing group *qsg-name*

Explanation

An attempt to add the queue manager specified by *qmgr-name* into a queue sharing group failed because the Db2 table specified by *table-name* indicates that the queue manager is already a member of a different queue sharing group identified by *qsg-name*.

System action

The utility program is terminated.

System programmer response

A queue manager can be a member of only one queue sharing group at any one time.

Either remove the queue manager from the queue sharing group it is in and resubmit the job or take no further action.

CSQU554E
QMGR *qmgr-name* entry cannot be removed from queue sharing group *qsg-name*, needed for structure *struc-name* backup

Explanation

The queue manager named by *qmgr-name* cannot be removed from the queue sharing group named by *qsg-name* because it has information about backups for structure *struc-name*. (The value shown for *struc-name* is the 12-character name as used by IBM MQ not the external name used by z/OS which includes the queue sharing group name.)

If the queue manager is needed for more than one structure, this message will be issued for each one.

System action

The utility program is terminated.

System programmer response

Using another queue manager in the queue sharing group, take a backup of the structure. Ensure that the EXCLINT time value used in the BACKUP CFSTRUCT command is less than the time since the queue manager that you are trying to remove was last stopped. Then resubmit the job.

When removing the last queue manager in a queue sharing group, you must use the FORCE option, rather than REMOVE. This removes the queue manager from the queue sharing group, whilst not performing the consistency checks of the queue manager logs being required for recovery. You should only perform this action if you are going to be deleting the queue sharing group; see Removing a queue manager from a queue sharing group for more information on managing queue sharing groups.

CSQU555E
QMGR *qmgr-name release* level is incompatible with queue sharing group *qsg-name* in Db2 table *table-name*

Explanation

An attempt to add the queue manager specified by *qmgr-name* into a queue sharing group failed because the Db2 table specified by *table-name* indicates that another queue manager in the queue sharing group is at an incompatible release level.

System action

The utility program is terminated.

System programmer response

Only queue managers with compatible release levels can be members of the same queue sharing group. For information about migration and compatibility between releases, see Maintaining and migrating.

CSQU556I
QSG *qsg-name* may contain unexpected characters

Explanation

The queue sharing group *qsg-name* being added specifies a queue sharing group name that either contains the '@' character, or is shorter than four characters and therefore has '@' characters appended to the short name to make the name four characters in length.

System action

Processing to add the queue sharing group continues. The utility will complete with return code 4.

System programmer response

Verify that the queue sharing group name specified by *qsg-name* is the intended name to be used for the queue sharing group. If not, use the utility to remove the queue sharing group, correct the queue sharing group name, and resubmit the request to add the queue sharing group.

The '@' character, although allowed in the *qsg-name*, is inadvisable as it is not supported as a character in an IBM MQ object name. Any definition such as queue manager alias definitions or other objects that need to refer to the *qsg-name*, will be unable to refer to the *qsg-name*. If possible, avoid using these characters.

CSQU557E
The QMGR and queue sharing group names must be different

Explanation

The attempt to add a queue manager to a queue sharing group failed because queue managers cannot have the same name as the queue sharing group to which they belong.

System action

The utility program is terminated.

CSQU558E
QMGR *qmgr-name* entry cannot be removed from queue sharing group *qsg-name*, SMDS for structure *struc-name* is not empty

Explanation

The queue manager named by *qmgr-name* cannot be removed from the queue sharing group named by *qsg-name* because it owns a shared message data set for structure *struc-name* which is not marked as empty, so it may still contain current message data. (The value shown for *struc-name* is the 12-character name as used by IBM MQ, not the external name used by z/OS which includes the queue sharing group name.)

System action

The utility program is terminated.

System programmer response

The queue manager cannot be removed until the owned shared message data set has been marked as empty, indicating that it has been closed normally by the owning queue manager at a time when it does not contain any message data. All shared messages with message data in the data set must have been read or marked as deleted first, and the owning queue manager must be connected to the structure in order to remove the deleted messages and free the data set space.

The current status of each shared message data set for the structure can be displayed using the command **DISPLAY CFSTATUS(struc-name) TYPE(SMDS)**.

CSQU559I
CSQ_ADMIN connection deleted for QMGR *qmgr-name*.

Explanation

Informational message indicating that a failed-persistent connection to the admin structure has been successfully deleted.

CSQU560I
Full name of admin structure is *admin-strname*

Explanation

This shows the full external name of the administration structure as used by z/OS, which includes the queue sharing group name.

CSQU561E
Unable to get attributes for admin structure, IXLMG RC=*rc* reason code=*reason*

Explanation

An attempt to add a queue manager to a queue sharing group failed; it was not possible to check the attributes of the administration structure because there was an XES IXLMG service error. The full name of the administration structure is given in the following CSQ570I message.

System action

The utility program terminates. The queue manager is not added to the queue sharing group.

System programmer response

Investigate the return and reason codes from the IXLMG service (both shown in hexadecimal), which are described in the *z/OS MVS Programming: Sysplex Services Reference* manual. If you are unable to resolve the problem, contact IBM Support.

CSQU562E
Admin structure attributes temporarily unavailable

Explanation

An attempt to add a queue manager to a queue sharing group failed; it was not possible to check the attributes of the administration structure because they were currently unavailable. The full name of the administration structure is given in the following CSQ570I message.

System action

The utility program terminates. The queue manager is not added to the queue sharing group.

System programmer response

Rerun the job later.

CSQU563I
Admin structure is defined in CF *cf-name*, allocated size *mm* KB, maximum entries *nn*

Explanation

This shows the current attributes of the administration structure for the queue sharing group. It is defined in the coupling facility named *cf-name*.

CSQU564E
Queue managers cannot be added to queue sharing group *qsg-name*, admin structure too small

Explanation

An attempt to add a queue manager to a queue sharing group failed; the current administration structure allocation is too small for a queue sharing group with the requested number of queue managers. The full name of the administration structure is given in the following CSQ570I message.

System action

The utility program terminates. The queue manager is not added to the queue sharing group.

System programmer response

See Specifying offload options for shared messages for information about coupling facility structure sizes for use with queue sharing groups.

The administration structure allocation must be increased before a new queue manager can be added to the queue sharing group. This may involve one or more of the following steps:

- Update the administration structure definition using the IXLMIAPU utility.
- Refresh the currently active CFRM policy.
- Dynamically alter the current allocation of the administration structure using the z/OS SETXCF START,ALTER command.

Rerun the job when the administration structure allocation has been increased.

CSQU565E
Unable to get attributes for admin structure, CF in failed state

Explanation

An attempt to add a queue manager to a queue sharing group failed; it was not possible to check the attributes of the administration structure because it is in a failed state. The full name of the administration structure is given in the following CSQ570I message.

System action

The utility program terminates. The queue manager is not added to the queue sharing group.

System programmer response

Use the z/OS DISPLAY XCF,STRUCTURE command to display the status of all structures in the currently active CFRM policy.

If the administration structure has failed, starting a queue manager in the queue sharing group will cause the structure to be allocated according to the current CFRM policy.

CSQU566I
Unable to get attributes for admin structure, CF not found or not allocated

Explanation

In attempting to add a queue manager to a queue sharing group, it was not possible to check the attributes of the administration structure because it has not yet been defined to the CFRM policy, or is not currently allocated in a coupling facility. The full name of the administration structure is given in the following CSQ570I message. If the structure is not allocated, then the structure will be allocated when the first queue manager starts.

System action

Processing continues.

System programmer response

Use the z/OS command DISPLAY XCF,STRUCTURE,STRNAME=<CFSTRNAME> to display the status (including size) of all structures in the currently active CFRM policy.

Ensure a structure definition exists in the CFRM policy. It will be needed before the queue manager can be started.

CSQU567E
QMGR *qmgr-name* not added to Db2 table due to a number mismatch.

Explanation

The QMGR qmgr-name could not be added to Db2 tables due to a mismatch in the QMGR numbers as indicated by message CSQU568E issued earlier.

System action

The utility terminates.

System programmer response

Add the QMGRs in the order corresponding to their QMGR number values in the XCF group, as can be seen by message CSQU524I when running CSQ5PQSG queue sharing group utility with the "VERIFY QSG" parameter.

If the issue is linked to a persistent failing connection to the CSQ_ADMIN structure, the problem can be resolved by clearing the CF structure using the SETXCF FORCE command.

CSQU568E
QMGR number mismatch for QMGR *qmgr-name* in queue sharing group *qsg-name*: DB2 value=*nn*, XCF member value=*nn*, CSQ_ADMIN connection value=*nn*, SMDS group value=*nn*

Explanation

The QMGR number is stored in the Db2 tables, the XCF group member, and the connections to the CF structures, and each stored value is displayed in the message. The QMGR number is created when a QMGR is added to a queue sharing group by using the queue sharing group utility (CSQ5PQSG).

This message indicates that there is a mismatch in the stored values for QMGR *qmgr-name* in queue sharing group *qsg-name* which will prevent the QMGR from starting.

System action

The utility terminates after all members in the XCF group have been processed.

System programmer response

If the QMGR number value is -1, the entry does not exist. Use the CSQ5PQSG utility with "ADD QMGR" parameter to add the missing entry.

If the QMGR number value is 0, the value has not been initialized (XCF group member and CSQ_ADMIN connection values only). Start the QMGR to initialize the value.

If the QMGR number value is greater than 0, collect the items listed in Coupling Facility codes (X'C5') and contact IBM Support.

If the SMDS Group number is -2, SMDS state information exists for the QMGR number for a different queue manager. Use the CSQ5PQSG utility with the VERIFY QMGR parameter to identify the queue manager associated with the SMDS state, followed by ADD QMGR to re-add the associated QMGR, then retry the original ADD QMGR command.

If the SMDS Group number is -1, SMDS state information exists for a queue manager with multiple qmids. Collect the items listed in Coupling Facility codes (X'C5') and contact IBM Support.

CSQU569E
Unexpected CSQ_ADMIN connection found for QMGR *qmgr-name*

Explanation

For each QMGR in the queue sharing group there should only be one connection to the CSQ_ADMIN structure. This message is issued for each additional connection found.

System action

The utility terminates after all members in the XCF group have been processed.

System programmer response

This situation should not occur. The connections can be displayed using the display XCF command for the CSQ_ADMIN structure.

Collect the items listed in Coupling Facility codes (X'C5') and contact IBM Support.

CSQU570I
QSG *qsg-name* successfully verified

Explanation

The request to verify information for queue sharing group *qsg-name* completed successfully. All the information is consistent.

CSQU571E
QSG *qsg-name* entry cannot be verified, not found in Db2 table *table-name*

Explanation

An attempt to verify the queue sharing group *qsg-name* failed because no entry for it was found in the Db2 table *table-name*.

System action

The utility program is terminated.

System programmer response

Ensure that the queue sharing group *qsg-name* was originally defined in the table *table-name*. Check that the utility job connected to the correct Db2 data-sharing group.

If necessary resubmit the job.

CSQU572E
Usage map *map-name* and Db2 table *table-name* inconsistent

Explanation

While verifying a queue sharing group, an inconsistency was found between the information in the usage map *map=name* and the Db2 table *table-name*. The following messages give more details about the inconsistency.

System action

Processing continues.

System programmer response

Check that the utility job connected to the correct Db2 data-sharing group. If necessary resubmit the job.

Contact IBM Support for assistance.

CSQU573E
QMGR *qmgr-name* in table entry *entry-number* not set in usage map

Explanation

While verifying a queue sharing group, an inconsistency was found between the information in a usage map and the corresponding Db2 table. The inconsistency is described in the message; preceding message CSQU572E identifies the usage map and table.

System action

Processing continues.

System programmer response

See message CSQU572E.

CSQU574E
QMGR *qmgr-name* in usage map has no entry in table

Explanation

While verifying a queue sharing group, an inconsistency was found between the information in a usage map and the corresponding Db2 table. The inconsistency is described in the message; preceding message CSQU572E identifies the usage map and table.

System action

Processing continues.

System programmer response

See message CSQU572E.

CSQU575E
Structure *struc-name* in table entry *entry-number* not set in usage map

Explanation

While verifying a queue sharing group, an inconsistency was found between the information in a usage map and the corresponding Db2 table. The inconsistency is described in the message; preceding message CSQU572E identifies the usage map and table. (The value shown for *struc-name* is the 12-character name as used by IBM MQ, not the external name used by z/OS which includes the queue sharing group name.)

System action

Processing continues.

System programmer response

See message CSQU572E.

CSQU576E
Structure *struc-name* in usage map has no entry in table

Explanation

While verifying a queue sharing group, an inconsistency was found between the information in a usage map and the corresponding Db2 table. The inconsistency is described in the message; preceding message CSQU572E identifies the usage map and table. (The value shown for *struc-name* is the 12-character name as used by IBM MQ, not the external name used by z/OS which includes the queue sharing group name.)

System action

Processing continues.

System programmer response

See message CSQU572E.

CSQU577E
Queue *q-name* in table entry *entry-number* not set in usage map for structure *struc-name*

Explanation

While verifying a queue sharing group, an inconsistency was found between the information in a usage map and the corresponding Db2 table. The inconsistency is described in the message; preceding message CSQU572E identifies the usage map and table. (The value shown for *struc-name* is the 12-character name as used by IBM MQ, not the external name used by z/OS which includes the queue sharing group name.)

System action

Processing continues.

System programmer response

See message CSQU572E.

CSQU578E
Queue *q-name* in usage map for structure *struc-name* has no entry in table

Explanation

While verifying a queue sharing group, an inconsistency was found between the information in a usage map and the corresponding Db2 table. The inconsistency is described in the message; preceding message CSQU572E identifies the usage map and table. (The value shown for *struc-name* is the 12-character name as used by IBM MQ, not the external name used by z/OS which includes the queue sharing group name.)

System action

Processing continues.

System programmer response

See message CSQU572E.

CSQU580I
DSG *dsg-name* is ready for migration

Explanation

The request to migrate the data-sharing group *dsg-name* to use new Db2 tables successfully verified that the data-sharing group is ready to be migrated.

System programmer response

Perform the migration.

CSQU581E
DSG *dsg-name* has incompatible QMGR levels in QSG *qsg-name*

Explanation

The data-sharing group *dsg-name* cannot be migrated to use new Db2 tables because the levels of the queue managers in queue sharing group *qsg-name*, which uses the data-sharing group, are incompatible.

System action

The utility program is terminated.

System programmer response

To perform the migration, all the queue managers in all the queue sharing groups that use the data-sharing group must have installed a PTF and been started, to bring them to the necessary level. Examine the CSQ.ADMIN_B_QMGR Db2 table to determine the levels of the queue managers and those which need to be upgraded. Look at the fields QMGRNAME, MVERSIONL, MVERSIONH and investigate the queue managers with the lower values in MVERSIONH.

For information about migration and compatibility between releases, see Maintaining and migrating.

CSQU582E
DSG *dsg-name* has already been migrated

Explanation

The data-sharing group *dsg-name* cannot be migrated to use new Db2 tables because it has already been migrated.

System action

The utility program is terminated.

System programmer response

As part of the migration, the CSQ.OBJ_B_CHANNEL Db2 table will have its row size increased above 4 KB. The utility found that such a row size slready exists. Examine the CSQ.OBJ_B_CHANNEL Db2 table to verify that the migration has already occurred.

For information about migration and compatibility between releases, see Maintaining and migrating.

CSQU583I
QSG *qsg-name* in DSG *dsg-name* is ready for migration

Explanation

The request to migrate the queue sharing group *qsg-name* in data-sharing group *dsg-name* to use new Db2 tables successfully verified that the queue sharing group is ready to be migrated.

System programmer response

Perform the migration. You should do this as a conditional step in the same job as the utility migration request, as shown in the sample jobs CSQ4570T and CSQ4571T in the SCSQPROC library.

CSQU584E
QSG *qsg-name* in DSG *dsg-name* has incompatible QMGR levels

Explanation

The queue sharing group *qsg-name* in data-sharing group *dsg-name* cannot be migrated to use new Db2 tables because the levels of the queue managers using the data-sharing group are incompatible.

System action

The utility program is terminated.

System programmer response

To perform the migration, all the queue managers in all the queue sharing groups that use the data-sharing group must have installed a PTF and been started, to bring them to the necessary level. Examine the CSQ.ADMIN_B_QMGR Db2 table to determine the levels of the queue managers and those which need to be upgraded.

For information about migration and compatibility between releases, see Maintaining and migrating.

CSQU585E

QSG *qsg-name* entry cannot be migrated, not found in Db2 table *table-name*

Explanation

The queue sharing group, *qsg-name*, cannot be migrated because no entry was found for it in the Db2 table, *table-name*.

System action

The utility program is terminated.

System programmer response

Ensure that the queue sharing group *qsg-name* was originally defined in the table *table-name*.

Check that the utility job is connected to the correct Db2 data-sharing group. If necessary resubmit the job.

CSQU586I

QMGR *qmgr-name* entry being removed from queue sharing group *qsg-name*, needed for structure *struc-name* backup

Explanation

The queue manager named by *qmgr-name* is being force removed from the queue sharing group named by *qsg-name* and it has information about backups for structure *struc-name*. (The value shown for *struc-name* is the 12-character name as used by IBM MQ, not the external name used by Db2 which includes the queue sharing group name.)

If the queue manager *qmgr-name* is added back to the queue sharing group it will cause an inconsistent state that could prevent structure *struct-name* from being recovered should it fail before a structure backup is taken.

If the queue manager is needed for more than one structure, this message is issued for each one.

System action

The queue manager is removed from the queue sharing group, and the utility program ends with return code 4.

System programmer response

If CF structure *struc-name* is usable, take a backup of CF structure *struc-name* as soon as possible using another queue manager in the queue sharing group. Otherwise, if the queue manager *qmgr-name* is added back to the queue sharing group it should be restarted before recovering structure *struc-name*.

CSQU587I

QMGR *qmgr-name* entry being removed from queue sharing group *qsg-name*, SMDS for structure *struc-name* is not empty

Explanation

The queue manager named by *qmgr-name* is being removed from the queue sharing group named by *qsg-name* while it owns a shared message data set for structure *struc-name* that is not marked as empty, so it may still contain current message data. (The value shown for *struc-name* is the 12-character name as used by IBM MQ, not the external name used by z/OS which includes the queue sharing group name.)

System action

The queue manager is removed from the queue sharing group, and the utility program ends with return code 4.

Messages on the SMDS for queue manager *qmgr-name* will remain accessible as long as the SMDS is retained.

CSQU599I

QSG *qsg-name* contains queue managers below release level *min-level*. The QSG does not support release level *current-level*

Explanation

The queue-sharing group contains queue managers below the minimum level supported by the release of IBM MQ for z/OS used to run CSQ5PQSG. You cannot start a queue manager at *current-level* in this queue-sharing group.

System programmer response

Locate the queue managers in the queue-sharing group that are below the *min-level* specified in the message and upgrade them to at least *min-level* before adding any *current-level* queue managers to the queue-sharing group.

CSQU680E

Db2 and CF structure out of sync for list header *list-header-number* in structure *struc-name*

Severity

8

Explanation

The row for the shared queue in Db2 represents a different queue than the one found in the CF structure for list *header list-header-number* in structure *struc-name*. This inconsistency causes the queue manager to abend with 5C6-00C51053 and issue message CSQE137E. Messages CSQU681I and CSQU682I are also issued, providing further details.

System action

The mismatch is reported and the utility continues processing.

System programmer response

Collect the items listed in Coupling Facility codes (X'C5') and Db2 manager codes (X'F5') and contact Db2 Support.

CSQU681I

Db2 entry for list header *list-header-number* in structure *struc-name: queue-name*

Severity

0

Explanation

This message is issued with message CSQU680E. *Queue-name* is the name of the queue found in Db2 for list header *list-header-number* in structure *struc-name*.

System action

The mismatch is reported and the utility continues processing.

System programmer response

Collect the items listed in Coupling Facility codes (X'C5') and Db2 manager codes (X'F5') and contact Db2 Support.

CSQU682I

CF entry for list header *list-header-number* in structure *struc-name: queue-name*

Severity

0

Explanation

This message is issued with message CSQU680E. *Queue-name* is the name of the queue found in the CF for list header *list-header-number* in structure *struc-name*.

System action

The mismatch is reported and the utility continues processing.

System programmer response

Collect the items listed in Coupling Facility codes (X'C5') and Db2 manager codes (X'F5') and contact Db2 Support.

CSQU683E

Missing CF entry for list header *list-header-number* in structure *struc-name*

Severity
8

Explanation

The Db2 entry for list header *list-header-number* in structure *struc-name* indicates that a current copy is available in the CF, however, the copy is not found. This inconsistency causes return code 2085 for applications trying to use this queue.

System action

The mismatch is reported and the utility continues processing.

System programmer response

Starting or restarting one of the queue managers in the queue sharing group will resolve the problem. If the problem persists, collect the items listed in Coupling Facility codes (X'C5') and in Db2 manager codes (X'F5') and contact IBM Support.

CSQU684I

Structure *struc-name* has not yet been allocated by a queue manager

Severity
0

Explanation

The CF structure *struc-name* is not allocated. This happens when the first **IXLCONN** to the structure is issued, and should only be issued by a queue manager in the QSG.

System action

The utility continues processing.

System programmer response

None.

CSQU685I

Structure *struc-name* connected

Severity
0

Explanation

The utility has successfully connected to CF structure *struc-name*.

System action

The utility continues processing.

System programmer response

None.

CSQU686E

Structure *struc-name* connection failed, **IXLCONN** RC=*return-code* reason=*reason*

Severity
8

Explanation

The utility failed to connect to CF structure *struc-name*.

System action

The utility skips any further queues for this structure and continues processing.

System programmer response

Examine the return and reason codes to determine why the **IXLCONN** connect command failed.

CSQU687I

Structure *struc-name* disconnected

Severity
0

Explanation

The utility has disconnected from CF structure *struc-name*.

System action

The utility continues processing.

System programmer response

None.

CSQU688E

Missing Db2 entry for list header *list-header-number* in structure *struc-name*

Severity
0

Explanation

The CF entry for list header *list-header-number* in structure *struc-name* indicates that a current copy is available in Db2, however, the copy is not found. This inconsistency causes a problem if a new queue is defined for the same list header.

System action

The mismatch is reported and the utility continues processing.

System programmer response

Collect the items listed in Coupling Facility codes (X'C5') and in Db2 manager codes (X'F5') and contact IBM Support.

CSQU689E

Unexpected return code for structure *struc-name*, **IXLLSTE** RC=*return-code* reason=*reason*

Severity
8

Explanation

The utility failed to read a list entry from the CF structure *struc-name*.

System action

The utility skips any further queues for this structure and continues processing.

System programmer response

Examine the return and reason codes to determine why the **IXLLSTE** read failed.

CSQU950I
csect-name IBM MQ for z/OS V*n*

Explanation

This is part of the header to the report issued by the utility program.

CSQU951I
csect-name Data Conversion Exit Utility - *date time*

Explanation

This is part of the header to the report issued by the utility program.

CSQU952I
csect-name Utility completed, return code=*ret-code*

Explanation

The utility completed. The return code is 0 if all the input was processed successfully, or 8 if any errors were found.

System action

The utility ends.

System programmer response

If the return code is non-zero, investigate the errors that were reported.

CSQU954I
n structures processed

Explanation

This indicates how many data structures were processed by the utility program.

CSQU956E
Line *line-number*: structure array field has incorrect dimension

Explanation

The dimension specified for a structure array field was not correct.

System action

Processing stops.

System programmer response

Correct the field specification and resubmit the job.

CSQU957E
Line *line-number*: structure has field following a variable length field

Explanation

There was an error in the indicated line. A variable length field must be the last field of a structure.

System action

Processing continues.

System programmer response

Correct the field specification and resubmit the job.

CSQU958E
Line *line-number*: structure field name has unsupported type 'float'

Explanation

There was an error in the indicated line. A field had a type of 'float', which is not supported.

System action

Processing continues.

System programmer response

Correct the field specification and resubmit the job, or provide your own routine for converting such fields.

CSQU959E
Line *line-number*: structure field name has unsupported type 'double'

Explanation

There was an error in the indicated line. A field had a type of 'double', which is not supported.

System action

Processing continues.

System programmer response

Correct the field specification and resubmit the job, or provide your own routine for converting such fields.

CSQU960E
Line *line-number*: structure field name has unsupported type 'pointer'

Explanation

There was an error in the indicated line. A field had a type of 'pointer', which is not supported.

System action

Processing continues.

System programmer response

Correct the field specification and resubmit the job, or provide your own routine for converting such fields.

CSQU961E
Line *line-number*: structure field name has unsupported type 'bit'

Explanation

There was an error in the indicated line. A field had a type of 'bit', which is not supported.

System action

Processing continues.

System programmer response

Correct the field specification and resubmit the job, or provide your own routine for converting such fields.

CSQU965E
Invalid EXEC PARM

Explanation

The EXEC PARM field was not blank.

System action

The utility is terminated.

System programmer response

Change the JCL, and resubmit the job.

CSQU968E
Unable to OPEN *ddname* data set

Explanation

The program was unable to open data set *ddname*.

System action

The utility is terminated.

System programmer response

Examine the error message that was sent to the job log to determine the reason for the error. Check that the data set was correctly specified.

CSQU970E
Line `line-number`: syntax error

Explanation

There was a syntax error in the indicated line.

System action

Processing stops.

System programmer response

Correct the error and resubmit the job.

CSQU971E
Unable to GET from *ddname* data set

Explanation

The program was unable to read a record from the *ddname* data set.

System action

The utility is terminated.

System programmer response

Examine the error message that was sent to the job log to determine the reason for the error. Check that the data set was correctly specified.

CSQU972E
Unable to PUT to *ddname* data set

Explanation

The program was unable to write the next record to the *ddname* data set.

System action

The utility is terminated.

System programmer response

Examine the error message that was sent to the job log to determine the reason for the error. Check that the data set was correctly specified.

CSQU999E
Unrecognized message code *ccc*

Explanation

An unexpected error message code was issued by the utility.

System action

Processing continues.

System programmer response

Note the code *ccc* (which is shown in hexadecimal) and contact IBM Support to report the problem.

z/OS *Agent services messages (CSQV...)*

CSQV086E
QUEUE MANAGER ABNORMAL TERMINATION REASON= *reason-code*

Explanation

The queue manager is ending abnormally, because an error that cannot be corrected has occurred. This message, which is not automatically deleted from the operator console, is issued during abnormal termination. *reason-code* is the termination reason code. If this abnormal termination is invoked multiple times, the termination reason code that accompanies this message is the reason associated with the first invocation.

System action

Abnormal termination processing continues.

System programmer response

For additional information, look up the reason code in "IBM MQ for z/OS codes" on page 943.

This message is accompanied by one or more dumps. Obtain a copy of SYS1.LOGREC after the queue manager completely terminates, and the dumps. If you suspect an error in IBM MQ, see Troubleshooting and support for information about identifying and reporting the problem.

CSQV400I
ARCHIVE LOG QUIESCE CURRENTLY ACTIVE

Explanation

An ARCHIVE LOG MODE(QUIESCE) command is currently active. This message is part of the DISPLAY LOG or DISPLAY THREAD command report.

System action

This message is issued as information only. It indicates that the ARCHIVE LOG MODE(QUIESCE) command has not completed and, consequently, updates against IBM MQ resources have been temporarily suspended. This might result in active threads being suspended awaiting termination of the quiesce period. Processing otherwise continues normally.

CSQV401I
DISPLAY THREAD REPORT FOLLOWS -

Explanation

This message is issued as the title for the DISPLAY THREAD command report output. It precedes the other messages generated by this command:

- Message CSQV402I provides the formatted report when the detailed status of active threads is requested using TYPE(ACTIVE).
- Message CSQV432I provides the formatted report when the summary status of active threads is requested using TYPE(REGIONS).
- Message CSQV406I provides the formatted report when the status of in-doubt threads is requested using TYPE(INDOUBT).
- Message CSQV436I provides the formatted report when the status of in-doubt threads on another queue manager is requested using TYPE(INDOUBT) with QMNAME.

System action

Processing continues normally.

CSQV402I
ACTIVE THREADS -

Explanation

This message is the response to the DISPLAY THREAD TYPE(ACTIVE) command. It provides the status information for each active thread, as follows:

```
  NAME S T REQ THREAD-XREF USERID ASID URID  name s t req thread-xref userid asid urid ⋮
DISPLAY ACTIVE REPORT COMPLETE
```

where:

name
The connection name, which is one of the following:

- z/OS batch job name
- TSO user ID
- CICS APPLID
- IMS region name
- Channel initiator job name

s
Connection status code:

N
The thread is in IDENTIFY status.

T
The thread has issued CREATE THREAD.

Q
The CREATE THREAD request has been queued. The associated allied task is placed in a wait state.

C
The thread is queued for termination as a result of the termination of the associated allied task. If this thread is also the last (or only) IBM MQ thread for the address space, the associated allied task is placed in a wait state.

D
The thread is in the process of termination as a result of the termination of the associated allied task. If this thread is also the last (or only) IBM MQ thread for the address space, the associated allied task is placed in a wait state.

An asterisk is appended if the thread is active within IBM MQ.

t
Connection type code:

B
Batch: From an application using a batch connection

R
RRS: From an RRS-coordinated application using a batch connection

C
CICS: From CICS

I
IMS: From IMS

S
System: From an internal function of the queue manager or from the channel initiator.

req
A wraparound counter to show the number of IBM MQ requests.

thread-xref
The recovery thread cross-reference identifier associated with the thread.

userid
The user ID associated with a connection. If not signed-on, this field is blank.

asid
A hexadecimal number representing the ASID of the home address space.

urid
Unit of recovery identifier. This is the log RBA of the current unit of recovery associated with the thread. If there is no current unit of recovery, it is shown as 0000000000000000.

Exceptionally, the last line might be:

```
DISPLAY ACTIVE TERMINATED WITH MAX LINES
```

if the report was generated in response to a command from a z/OS console and more than 252 response messages were generated. Only 252 response messages are returned.

System action

Processing continues normally.

CSQV406I
INDOUBT THREADS -

Explanation

This message is the response to the DISPLAY THREAD TYPE(INDOUBT) command. It provides the status information for each in-doubt thread, as follows:

```
NAME THREAD-XREF URID NID  name thread-xref urid origin-id ⋮  DISPLAY INDOUBT REPORT
COMPLETE
```

where:

name
The connection name, which is one of the following:

- z/OS batch job name
- TSO user ID
- CICS APPLID
- IMS region name
- Channel initiator job name

thread-xref
The recovery thread cross-reference identifier associated with the thread. See Connecting from the IMS control region for more information.

urid
Unit of recovery identifier. This is the log RBA of the current unit of recovery associated with the thread. (This is omitted if the command was issued from a z/OS console with a non-specific connection name.)

origin-id
The origin identifier, a unique token identifying the unit of recovery within the queue manager. This has the form *origin-node*.*origin-urid*, where:

origin-node
A name that identifies the originator of the thread. (This is omitted for batch RRS connections.)

origin-urid
The hexadecimal number assigned to the unit of recovery for this thread by the originating system.

Exceptionally, the last line might be:

```
DISPLAY INDOUBT TERMINATED WITH MAX LINES
```

if the report was generated in response to a command from a z/OS console and more than 252 in-doubt threads were eligible for display.

System action

Processing continues normally.

CSQV410I
NO ACTIVE CONNECTION FOUND FOR NAME=*connection-name*

Explanation

The DISPLAY THREAD command was unable to find any active connection associated with *connection-name*.

System action

Command processing continues.

CSQV411I
NO ACTIVE THREADS FOUND FOR NAME=*connection-name*

Explanation

The DISPLAY THREAD command was unable to locate any active threads associated with *connection-name*.

System action

Command processing continues.

CSQV412I
csect-name NO INDOUBT THREADS FOUND FOR NAME=`connection name`

Explanation

The DISPLAY THREAD command was unable to locate any in-doubt threads associated with *connection name*.

System action

Command processing continues.

CSQV413E
csect-name CONNECTION NAME MISSING

Explanation

A connection name was not supplied with the command, and a default connection name cannot be determined.

System action

Command processing terminates.

CSQV414I
THREAD NID=*origin-id* COMMIT SCHEDULED

Explanation

The thread specified by the recovery origin identifier *origin-id* is scheduled for COMMIT recovery action.

System action

Processing continues.

CSQV415I
THREAD NID=*origin-id* BACKOUT SCHEDULED

Explanation

The thread specified by the recovery origin identifier *origin-id* is scheduled for BACKOUT recovery action.

System action

Processing continues.

CSQV416E
THREAD NID=*origin-id* IS INVALID

Explanation

The RESOLVE INDOUBT command determined that the input format for the specified thread *origin-id* is invalid.

System action

Command processing continues.

CSQV417I
THREAD NID=*origin-id* NOT FOUND

Explanation

The RESOLVE INDOUBT command was unable to locate the thread specified by the recovery origin identifier *origin-id* to be scheduled for recovery. Either the thread identifier is incorrect, or the thread is no longer in an in-doubt state.

System action

Command processing continues.

CSQV419I
NO ACTIVE CONNECTIONS FOUND

Explanation

A DISPLAY THREAD(*) TYPE(ACTIVE) or TYPE(REGIONS) command was issued for all threads, but no active connections were found.

System action

Command processing continues.

CSQV420I
NO INDOUBT THREADS FOUND

Explanation

A DISPLAY THREAD(*) TYPE(INDOUBT) command was issued for all threads, but no in-doubt threads were found.

System action

Command processing continues.

CSQV423I
cmd MESSAGE POOL SIZE EXCEEDED

Explanation

The storage requirement needed to generate responses for the command *cmd* exceeded the maximum size of the message buffer pool.

System action

Processing is terminated.

CSQV424I
THREAD ID=*thread-xref* COMMIT SCHEDULED

Explanation

The thread specified by the recovery thread cross-reference identifier *thread-xref* is scheduled for COMMIT recovery action.

System action

Processing continues.

CSQV425I
THREAD ID=*thread-xref* BACKOUT SCHEDULED

Explanation

The thread specified by the recovery thread cross-reference identifier *thread-xref* is scheduled for BACKOUT recovery action.

System action

Processing continues.

CSQV427I
THREAD ID=*thread-xref* NOT FOUND

Explanation

The RESOLVE INDOUBT command was unable to locate the thread specified by the recovery thread cross-reference identifier *thread-xref* to be scheduled for recovery. Either the thread identifier is incorrect, or the thread is no longer in an in-doubt state.

System action

Command processing continues.

CSQV428I
CURRENT THREAD LIMIT OF *nnn* EXCEEDED. CREATE THREAD FOR JOB *jobname* DEFERRED

Explanation

A job requested a connection to the queue manager, but the current number of connections is the maximum allowed.

System action

The request for a connection is suspended, and waits until another connection ends.

System programmer response

If this situation occurs frequently, contact IBM Support for assistance.

CSQV432I
ACTIVE THREADS -

Explanation

This message is the response to the DISPLAY THREAD TYPE(REGIONS) command. It provides the status information for each active connection, as follows:

```
  NAME TYPE USERID ASID THREADS  name type userid asid threads ⋮  DISPLAY ACTIVE REPORT
COMPLETE
```

where:

name
The connection name, which is one of the following:

- z/OS batch job name
- TSO user ID
- CICS APPLID
- IMS region name

- Channel initiator job name

type
The connection type:

CICS
From CICS.

IMS
From IMS.

BATCH
From an application using a batch connection.

RRSBATCH
From an RRS-coordinated application using a batch connection.

CHINIT
From the channel initiator.

userid
The user ID associated with a connection. If not signed-on, this field is blank.

asid
A hexadecimal number representing the ASID of the home address space.

threads
The number of active threads associated with the connection. This excludes fixed internal threads, such as those for the CICS adapter tasks, or the channel initiator listeners.

Exceptionally, the last line might be:

```
DISPLAY ACTIVE TERMINATED WITH MAX LINES
```

if the report was generated in response to a command from a z/OS console and more than 252 response messages were generated. Only 252 response messages are returned.

System action

Processing continues normally.

CSQV433I
'QMNAME' NOT ALLOWED, NOT IN QUEUE SHARING GROUP

Explanation

A DISPLAY THREAD TYPE(INDOUBT) or RESOLVE INDOUBT command specifying the QMNAME keyword was issued, but the requesting queue manager *qmgr-name* is not in a queue sharing group or the requested queue manager *qmgr-name* is not a member of the queue sharing group.

System action

Processing for the command is terminated.

CSQV434E
'QMNAME' ALLOWED ONLY WITH TYPE(INDOUBT)

Explanation

A DISPLAY THREAD command specifying the QMNAME keyword was issued, but TYPE(INDOUBT) was not specified.

System action

Processing for the command is terminated.

CSQV435I
QMNAME(qmgr-name) IS ACTIVE, COMMAND IGNORED

Explanation

A DISPLAY THREAD TYPE(INDOUBT) or RESOLVE INDOUBT command specifying the QMNAME keyword was issued, but the requested queue manager *qmgr-name* is active.

System action

Processing for the command is terminated.

CSQV436I
INDOUBT THREADS FOR *qmgr-name* -

Explanation

This message comprises the response to the DISPLAY THREAD TYPE(INDOUBT) command when the QMNAME keyword was specified. It provides the status information for each in-doubt unit-of-work on the requested queue manager; the information is displayed in the same format as in message CSQV406I.

System action

Processing continues normally.

CSQV437I
CANNOT RESOLVE THREAD NID=*origin-id*, SOME RESOURCES UNAVAILABLE

Explanation

The RESOLVE INDOUBT command was unable to schedule the thread specified by the recovery origin identifier *origin-id* for recovery, because not all the resources necessary for recovery were available.

System action

The identified thread will remain in-doubt.

CSQV450I
csect-name Unable to open *ddname* data set

Explanation

The *ddname* data set could not be opened, as reported in the preceding messages.

System action

Processing continues, but functions that require the data set will be inhibited.

System programmer response

Investigate the problem reported in the preceding messages.

CSQV451I
csect-name Unable to get storage for exits, RC=*return-code*

Explanation

An attempt to obtain some storage for use by exits failed. *return-code* is the return code (in hexadecimal) from the z/OS STORAGE service.

System action

Processing continues, but cluster workload user exits will not be available.

System programmer response

Go to the appropriate volume of the *z/OS MVS Programming: Assembler Services Reference* documentation for information about the return code from the STORAGE request.

CSQV452I
csect-name Cluster workload exits not available

Explanation

Cluster workload user exit functions will not be available, because:

- There is no CSQXLIB DD statement in the started task JCL procedure for the queue manager, xxxxCHIN
- The EXITTCB system parameter is zero.

System action

Processing continues, but cluster workload user exits will not be available.

System programmer response

If you want to use cluster workload exits, add the required statement to the queue manager started task JCL procedure and specify a non-zero value for the EXITTCB system parameter. For more information about cluster workload exits, see Cluster workload exit programming.

CSQV453I
csect-name Unable to load *module-name*, reason=*ssssrrrr*

Explanation

The queue manager was unable to load a module required for exits. *ssss* is the completion code and *rrrr* is the reason code (both in hexadecimal) from the z/OS LOAD service.

System action

Processing continues, but cluster workload user exits will not be available.

System programmer response

Check the console for messages indicating why the module was not loaded.

Go to the appropriate volume of the *z/OS MVS Programming: Assembler Services Reference* documentation for information about the codes from the LOAD request.

Ensure that the module is in the required library, and that it is referenced correctly. The queue manager attempts to load this module from the library data sets under the STEPLIB DD statement of its started task JCL procedure xxxxMSTR.

CSQV455E
csect-name Cluster workload exit `exit-name` timed out

Explanation

A cluster workload user exit did not return to the queue manager within the allowed time, as specified by the EXITLIM system parameter.

System action

The exit is disabled until its load module in the CSQXLIB data set is refreshed.

System programmer response

Investigate why your exit is not returning in time.

CSQV456E
csect-name Cluster workload exit error, TCB=*tcb-name* reason=*sssuuu-reason*

Explanation

The exit subtask using TCB *tcb-name* is ending abnormally because an error that cannot be corrected has occurred in a cluster workload user exit. *sss* is the system completion code, *uuu* is the user completion code, and *reason* is the associated reason code (all in hexadecimal).

System action

The subtask ends abnormally, and a dump is normally issued. The exit is disabled until its load module in the CSQXLIB data set is refreshed.

System programmer response

User completion codes are generally the result of errors detected by the exit itself. If a system completion code is shown, see the *z/OS MVS System Codes* documentation for information about the problem in your exit.

CSQV457E

csect-name Unable to establish ESTAE, RC=*return-code*

Severity

8

Explanation

During startup processing, the recovery environment for a cluster workload user exit task could not be set up. *return-code* is the return code (in hexadecimal) from the z/OS ESTAE service.

System action

The task does not start. Cluster workload user exits will be available providing at least one task starts.

System programmer response

Go to the appropriate volume of the *z/OS MVS Programming: Assembler Services Reference* documentation for information about the return code from the ESTAE request. If you are unable to solve the problem, contact IBM Support for assistance.

CSQV459I

csect-name Unable to free storage for exits, RC=*return-code*

Explanation

An attempt to release some storage that was used by exits failed. *return-code* is the return code (in hexadecimal) from the z/OS STORAGE service.

System action

Processing continues.

System programmer response

Go to the appropriate volume of the *z/OS MVS Programming: Assembler Services Reference* documentation for information about the return code from the STORAGE request.

CSQV460I

csect-name Cluster workload exits are disabled but **CLWLEXIT** is set.

Explanation

A **CLWLEXIT** value is on the queue manager when it is started, however cluster workload exits are not enabled, and are prevented from operating.

System action

Message CSQV461D is issued, but **CLWLEXITs** are not enabled.

CSQV461D

csect-name

Reply Y to continue startup with CLWLEXIT not enabled, or N to shutdown.

Explanation

Issued after message CSQV460I. Due to a **CLWLEXIT** being set in the queue manager with cluster workload exits not enabled, a reply is required to continue startup with cluster workload exits not enabled.

System action

Queue manager startup waits for the reply from the operator. Replying Y allows the queue manager to continue startup with cluster workload exists not enabled. Replying N shuts down the queue manager with abend reason 00D40039.

While cluster workload exits are not enabled, the **CLWLEXIT** value can only be changed to a blank value, and the exit will not function. You should use the IBM MQ supplied workload balancing algorithm and attributes, to alter how objects are selected, and remove the **CLWLEXIT** value. See Workload balancing in clusters for more information.

For further information contact IBM Support.

z/OS *Instrumentation facilities messages (CSQW...)*

CSQW001I
ASYNCHRONOUSLY GATHERED DATA IS BEING FORMATTED

Explanation

The dump formatting exit is not using summary dump records for formatting. The formatted control blocks might not contain the same values as they did at the time of the error.

System action

Dump formatting continues.

System programmer response

If you want summary dump records to be used, do not specify the 'SUMDUMP=NO' operand on the MQ DUMP DISPLAY MAIN MENU (if you are using the dump display panels), or in the CSQWDMP verbexit (if you are using line mode IPCS).

CSQW002I
SUMMARY DUMP RECORDS ARE BEING FORMATTED

Explanation

The dump formatting exit is using MQ summary dump record information to format its control blocks.

System action

Dump formatting continues.

System programmer response

If you do not want IBM MQ summary dump records to be used in formatting, specify the 'SUMDUMP=NO' and 'SUBSYS=subsystem name' on the MQ DUMP DISPLAY MAIN MENU (if you are using the dump display panels), or in the CSQWDMP verbexit (if you are using line mode IPCS). Both operands are required.

CSQW004E
ONE OR MORE OPERANDS ARE NOT VALID. FORMATTING TERMINATED

Explanation

An invalid operand was specified on the MQ DUMP DISPLAY MAIN MENU (if you are using the dump display panels), or in the CSQWDMP verbexit (if you are using line mode IPCS).

System action

The dump formatting exit terminates.

System programmer response

Correct the operand specified by message CSQW007E.

CSQW006E
THE ERLY BLOCK CANNOT BE ACCESSED OR IT IS INVALID

Explanation

The dump formatting exit could not locate its anchor block.

System action

The dump formatting exit terminates.

System programmer response

Specify 'SUBSYS=subsystem name', and 'SUMDUMP=NO' on the MQ DUMP DISPLAY MAIN MENU (if you are using the dump display panels), or in the CSQWDMP verbexit if you are using line mode IPCS.

CSQW007E
OPERAND IS NOT VALID: *xxxx*

Explanation

The specified operand was not a valid dump formatting operand.

System action

The dump formatting exit terminates.

System programmer response

Check the dump formatting operands.

CSQW008E
THE SCOM CANNOT BE ACCESSED OR IT IS INVALID

Explanation

An error was encountered while trying to retrieve the SCOM.

System action

The dump formatting exit terminates.

System programmer response

If 'SUMDUMP=NO' was specified on the MQ DUMP DISPLAY MAIN MENU (if you are using the dump display panels), or in the CSQWDMP verbexit (if you are using line mode IPCS) omit it and resubmit the request. Otherwise, specify this operand, and resubmit the request.

CSQW009E
THE ADDRESS SPACE REQUESTED IS NOT AVAILABLE

Explanation

The MQ control blocks for the address space specified could not be located.

System action

Formatting continues of any other requested dump segment.

System programmer response

Check the ASID specified. The ASID must be specified in hexadecimal.

CSQW010E
THE TRACE RMFT CANNOT BE ACCESSED OR IT IS INVALID

Explanation

The MQ trace table could not be located.

System action

Formatting of the MQ trace table is bypassed, and formatting continues of any other requested dump segment.

System programmer response

If 'SUMDUMP=NO' was specified try formatting the dump again using the summary dump because it could contain the information required to access this data.

If 'SUMDUMP=NO' was not specified, and the summary dump was used, try formatting the dump again specifying this option because the summary dump data could have been corrupted.

CSQW011I
A LARGER REGION SIZE IS REQUIRED FOR THIS JOB

Explanation

The dump formatting exit could not obtain a large enough work buffer to process the summary dump records.

System action

The dump formatting exit terminates.

System programmer response

Rerun the job, specifying a larger TSO region size (or a larger region size if running in batch).

CSQW013I
DMPW NOT FOUND IN SUMMARY DUMP

Explanation

The dump formatting exit was unable to locate the DMPW control block in the summary record portion of the dump data set. Because the DMPW provides the main anchor block for the dump formatter, processing is terminated.

System action

The dump formatting exit terminates.

System programmer response

Specify 'SUBSYS=xxxx' to identify which address space to format information for.

CSQW014I
REQUIRED SUMMARY DUMP RECORDS ARE NOT IN THIS DUMP. WILL ATTEMPT TO FORMAT FROM NON-SUMMARY DUMP

Explanation

Expected data could not be found in the summary dump. This message is issued for information only. Dump formatting continues.

System action

Formatting is attempted using information found from the full dump instead of the summary dump.

CSQW015I
SSCVT NOT LOCATED, CHECK THE SUBSYSTEM NAME SPECIFIED

Explanation

In a search through the SSCVT chain, a match of the subsystem name in the SSCVTs and the subsystem name specified was not found.

System action

Formatting for the named subsystem is not done.

System programmer response

Specify the subsystem name correctly.

CSQW016I
THE RMVT CANNOT BE ACCESSED OR IT IS INVALID

Explanation

The dump formatting exit could not locate the RMVT. The RMVT is required for formatting the MQ trace table and a number of other MQ control blocks.

System action

Formatting of the MQ trace table is bypassed, and formatting of other requested dump segments continues.

System programmer response

If 'SUMDUMP=NO' was specified try formatting the dump again using the summary dump because it could contain the information required to access this data.

If 'SUMDUMP=NO' was not specified, and the summary dump was used, try formatting the dump again specifying this option because the summary dump data could have been corrupted.

CSQW017E
MAXIMUM STACK LEVEL EXCEEDED

Explanation

This condition is usually caused by the MQ control block formatter looping. The stack array is depleted and can no longer accommodate control blocks.

System action

Dump formatting is terminated.

System programmer response

Contact IBM Support.

CSQW018I
SUBSYS= SPECIFIED INCORRECTLY OR MISSING. REQUIRED IF SUMDUMP=NO SPECIFIED

Explanation

The 'SUMDUMP=NO' option was specified, but either the 'SUBSYS=' operand is missing, or it was incorrectly specified.

System action

Dump formatting is terminated.

System programmer response

Specify the name of the subsystem in the 'SUBSYS=' operand, and resubmit the request.

CSQW020I
UNSUCCESSFUL SEARCH FOR THE ERLY CONTROL BLOCK

Explanation

A key control block could not be located in the dump.

System action

Dump formatting is terminated.

System programmer response

Check that the 'SUBSYS=' operand was correctly specified, and resubmit the request.

CSQW022I
THE RESIDENT TRACE WAS NOT ACTIVE AT THE TIME OF DUMP

Explanation

Trace table formatting has been attempted, but no trace table existed at the time of the dump.

System action

Dump formatting continues with any other control blocks that were to be formatted.

CSQW023I
THE TRACE TABLE ENTRY IS OUT OF SEQUENCE OR OVERLAID

Explanation

A trace entry is overlaid by another trace entry of a different time stamp. This message is issued to flag an unrecognized trace entry. This error can occur if the dump is initiated by operator command, because the MQ address space continues to run while the dump is being taken.

System action

Formatting of the trace table continues.

CSQW024I
TRACE TABLE

Explanation

This identifies the start of the formatted trace table.

System action

Trace table formatting follows.

CSQW025I
ERROR ACCESSING THE TRACE TABLE

Explanation

A nonzero return code was returned from the storage access routine when accessing the trace table.

System action

Trace table formatting is bypassed.

CSQW026I
CONTROL BLOCK SUMMARY (ALL ADDRESS SPACES)

Explanation

This messages provides descriptive information about the type of formatting being produced.

System action

Dump formatting continues.

CSQW027I
CONTROL BLOCK SUMMARY (SINGLE ADDRESS SPACE)

Explanation

This messages provides descriptive information about the type of formatting being produced.

System action

Dump formatting continues.

CSQW028I
CONTROL BLOCK SUMMARY (LONG FORM GLOBAL)

Explanation

This messages provides descriptive information about the type of formatting being produced.

System action

Dump formatting continues.

CSQW029I
CONTROL BLOCK SUMMARY (SHORT FORM GLOBAL)

Explanation

This messages provides descriptive information about the type of formatting being produced.

System action

Dump formatting continues.

CSQW030E
DUMP ACCESS ERROR ACCESSING THE CONTROL BLOCK STRUCTURE TABLE IN THE DUMP

Explanation

A control block identifying the structure of MQ control blocks could not be found.

System action

Control block formatting is terminated.

System programmer response

Check the z/OS console to see if any messages were produced to indicate that there was a problem when the dump was taken. If you suspect an error in IBM MQ, see Troubleshooting and support for information about reporting the problem.

CSQW032E
ERROR ACCESSING ANCHOR CONTROL BLOCK

Explanation

A control block cannot be accessed from the dump.

System action

Control block formatting is terminated.

System programmer response

Check the z/OS console to see if any messages were produced to indicate that there was a problem when the dump was taken. If you suspect an error in IBM MQ, see Troubleshooting and support for information about reporting the problem.

CSQW033I
BEGINNING FORMATTING

Explanation

Formatting of MQ control blocks is beginning.

CSQW034I
TRACE TABLE AND GLOBAL BLOCKS ALREADY FORMATTED

Explanation

An indicative dump is being requested. The MQ trace table and the global blocks have already been formatted with first dump (full dump) for this abend dump (SNAP) invocation. These are, therefore, not formatted for this task.

CSQW035I
WARNING - NO TASK RELATED CONTROL BLOCKS FOR THIS TASK

Explanation

The task for which the dump is being requested is not identified to MQ. Task-related control blocks are not dumped. The MQ trace table and global blocks are dumped only if the SYSABEND DD statement is present and only if this is the first of the dumps (full dump) for this abend dump (SNAP) invocation.

System action

No MQ formatting is done for the specified task.

CSQW036I
CONTROL BLOCKS FOR TASKS ASSOCIATED WITH THE ABOVE RECOVERY COORDINATOR TASK

Explanation

The formatted blocks following this message are associated with tasks that have been identified to MQ with the 'recovery coordinator = no' option. These tasks might not have invoked SNAP, but they are associated with the task that did.

System action

The appropriate control blocks are formatted.

System programmer response

Examine the control blocks for relevant information.

CSQW037I
TASK RELATED CONTROL BLOCKS FOR THIS TASK

Explanation

The formatted blocks following this message are associated with the current task.

System action

The appropriate control blocks are formatted.

System programmer response

Examine the control blocks for relevant information.

CSQW038I
END OF FORMATTING

Explanation

Formatting of MQ control blocks is completed.

CSQW039I
FORMATTING COMPLETE FOR THIS DUMP

Explanation

The dump formatting exit has completed its processing for this dump data set.

CSQW041E
THE TAB CANNOT BE ACCESSED OR IT IS INVALID

Explanation

The MQ trace table anchor block could not be located.

System action

Formatting of the MQ trace table is bypassed, and formatting of any other requested dump segment continues.

System programmer response

If 'SUMDUMP=NO' was specified try formatting the dump again using the summary dump because it could contain the information required to access this data.

If 'SUMDUMP=NO' was not specified, and the summary dump was used, try formatting the dump again specifying this option because the summary dump data could have been corrupted.

Check the z/OS console to see if any messages were produced to indicate that there was a problem when the dump was taken. If you suspect an error in IBM MQ, see Troubleshooting and support for information about reporting the problem.

CSQW042E
REQUIRED SUMMARY DUMP RECORDS ARE NOT IN THIS DUMP. RERUN SPECIFYING SUBSYS= PARAMETER

Explanation

The summary dump records were not found in the dump. When this occurs the dump formatter needs the subsystem name to be able to identify which address space is to be formatted.

System action

Dump formatting is terminated.

System programmer response

Rerun the formatting specifying the parameter the subsystem name (using 'SUBSYS=').

CSQW049I
OLDEST SLOT ADDRESS INVALID, FORMATTING TRACE TABLE FROM FIRST ENTRY

Explanation

There are several pointers in the control block that defines the trace. One points to the start of the storage that contains the trace data, one to the end, and one to the next free record. The formatter has detected that the pointer to the next free record is outside the range indicated by the pointers to the start and end of the storage.

System action

Dump formatting continues, but from the physical start of the trace table, not the oldest record.

System programmer response

If the time of day values are meaningful, and in sequence, scan down the formatted trace to find the latest trace record written.

CSQW050I
ssnm NO SDWA/LOGREC, ABN=*comp-reason*, U=*userid*, M=*module*, C=*compid.vrm.comp-function*

Explanation

This message provides the default SVC dump title (SDUMP) associated with the SYS1.DUMP data set, when an SDWA was unavailable during recovery processing. The individual variable fields contain:

Field
Contents

ssnm
MQ subsystem name

ABN
The abend completion code, followed by the abend reason code

U
The user ID for the individual subsystem user

M
The function recovery routine responsible for the dump

C
The component-ID

vrm
The MQ version, release number, and modification level

comp-function
The component-ID function

System action

Dump processing continues.

System programmer response

Since the SDWA provides important diagnostic information to assist in problem determination, the recovery environment at time of error should be examined to determine why an SDWA was not provided for this ABEND.

In a non-recovery environment, there might be valid reasons for the lack of an SDWA (for example, the operator could have initiated the dump).

CSQW051E
ERROR DURING DUMP PROCESSING

Explanation

This message is generated by the recovery routine of the SDUMP dump data gathering service when an error is encountered during dump processing.

System action

Processing of the SUMLSTA user storage areas is terminated, an SVC dump is requested, and control is returned to RTM.

System programmer response

This error is documented in a SYS1.LOGREC record. This message can be issued because of an error in the invocation of SDUMP, or because of an error in SDUMP itself, or during control block examination and access.

CSQW053I
VRA DIAGNOSTIC INFORMATION REPORT

Explanation

The variable recording area (VRA) is part of the system diagnostic work area (SDWA) and contains MQ diagnostic information. The VRA is extracted and displayed in this report.

For information about this report, see Troubleshooting and support .

System action

Dump formatting continues.

CSQW054I
NO VRA DATA RECORDED IN SDWA

Explanation

The SDWA obtained from the SYS1.DUMP data set contained no diagnostic information in the VRA.

System action

VRA report generation is bypassed, dump format processing continues.

CSQW055I
UNABLE TO LOCATE SDWA

Explanation

The z/OS summary dump data access service routine (IEAVTFRD) was unable to locate the SDWA in the summary data portion of the SYS1.DUMP data set. SVC dumps only contain an SDWA if they are initiated by MQ. If the dump was initiated by any other means (such as the operator) the SDWA will not be present.

System action

No VRA is produced, and dump formatting continues.

CSQW056I
VRA DIAGNOSTIC REPORT COMPLETE

Explanation

The dump formatter has completed processing of the VRA diagnostic report.

System action

Dump formatting continues.

CSQW059I
SUMMARY OF CONNECTED JOBS

Explanation

A summary of information about connected jobs follows.

System action

Job summary information follows.

CSQW060I
BEGIN SAVE AREA TRACE

Explanation

This message identifies the start of the MQ register save area trace report which appears in the formatted section of an MQ SVC dump. This report is useful for problem determination because it contains the save areas for the agent execution block (EB) in error, and all associated agent EBs, traced from the point of error and displayed in order of invocation.

System action

Save area trace format processing continues for the agent EB in error, and all associated agent EBs.

CSQW061I
SAVE AREA TRACE COMPLETE

Explanation

This message indicates that the MQ formatted save area trace report (CSQW060I) is complete.

System action

Dump formatting continues.

CSQW062I
R6 (*R6-contents*) DOES NOT CONTAIN A VALID EB ADDRESS

Explanation

During dump format processing of the MQ formatted save area trace report (CSQW060I), register 6 (R6) did not contain the address of a valid agent execution block (EB).

System action

Save area trace format processing is terminated for the current agent EB, and all prior EBs.

CSQW063E
name (*address*) ASID (*asid*) NOT FOUND IN DUMP

Explanation

During processing of the save area trace report (CSQW060I), a control block or save area was not found in the dump data set.

Because the dump formatter uses the MQ and z/OS control blocks defined under the *name* field of this message to locate individual register save areas, subsequent save areas located using the *named* control block or save area will not be displayed in the report.

name
Identifies the name of the control block or save area that was not found in the dump data set:

SA
Indicates a save area

ASCE
MQ address space control element

EB
MQ execution block

TCB
z/OS task control block

RB
z/OS request block

XSB
z/OS extended status block

PSA
z/OS prefix save area

SDWA
z/OS system diagnostic work area

STSV
z/OS SRB status save area

STKE
z/OS cross memory stack element

address
The address of the named control block or save area.

asid
The address space identifier associated with the control block or save area.

Due to the execution structures and environmental restrictions of selected MQ and z/OS control structures, some control blocks and save areas associated with these execution environments will not be included in the dump data set.

System action

Register save area trace format processing for the current save area chains is terminated. Subsequent save area processing will vary depending on the specific control block or save area that was available, and the MQ agent execution environments at the time of the error.

CSQW064I
ERROR BLOCK NOT FOUND IN DUMP

Explanation

The dump formatter was unable to format a control block because the storage could not be found.

System action

Dump formatting continues.

CSQW065I
ERROR BLOCK LENGTH INCORRECT

Explanation

During the formatting of a control block, a mismatch was found between the expected length and the value determined from the dump.

System programmer response

You might find this message helpful when solving a more serious problem because it might indicate that a control block has been corrupted.

CSQW066I
ERROR BLOCK ID INCORRECT

Explanation

Each control block type has a unique identifier for verification. During the formatting of the control block, a mismatch occurred between the value expected and the value found in the control block in the dump.

System programmer response

This message could indicate that storage has been overlaid, and you might find it helpful when solving a more serious problem because it might indicate that a control block has been corrupted.

CSQW067I
ERROR BLOCK CHAINED FROM THIS BLOCK NOT FOUND IN DUMP

Explanation

Control blocks can contain pointers to other control blocks. A control block pointed to by the current control block could not be found in the dump.

System programmer response

This message could indicate that storage has been overlaid, and you might find it helpful when solving a more serious problem. The control block pointed to will have error message CSQW064I associated with it.

CSQW068I
ERROR BLOCK CHAINED FROM THIS BLOCK HAS INCORRECT ID

Explanation

Each control block type has a unique identifier for verification. During the formatting of a control block pointed to by the current control block, a mismatch occurred between the value expected and the value found in the control block in the dump.

System programmer response

This message could indicate that storage has been overlaid, and you might find it helpful when solving a more serious problem because it might indicate that a control block has been corrupted. The control block in error has error message CSQW066I associated with it.

CSQW069I
ERROR BLOCK EYECATCHER INCORRECT

Explanation

Each control block type has a unique eyecatcher for verification. During the formatting of the control block, a mismatch occurred between the value expected and the value found in the control block in the dump.

System programmer response

This message could indicate that storage has been overlaid, and you might find it helpful when solving a more serious problem because it might indicate that a control block has been corrupted.

CSQW070I
DUMP TITLE *dump-title*

Explanation

This shows the title of the dump.

CSQW072I
ENTRY: MQ user parameter trace

Explanation

This message is inserted into the formatted MQ trace to indicate that the control block was traced on entry to MQ.

CSQW073I
EXIT: MQ user parameter trace

Explanation

This message is inserted into the formatted MQ trace to indicate that the control block was traced on exit from MQ.

CSQW074I
ERROR: MQ user parameter trace

Explanation

This message is inserted into the formatted MQ trace to indicate that the control block was traced because it was determined to be in error.

CSQW075I
WARNING - data was truncated at 256 bytes

Explanation

This message is inserted into the formatted MQ trace when a control block has exceeded a 256 byte length limit.

CSQW076I
Return code was *mqrc*

Explanation

This message is inserted into the formatted MQ trace when an error has been detected. *mqrc* is the return code. Refer to "API completion and reason codes" on page 1151 for information about this code.

CSQW105E
ERROR DURING LOAD OR VALIDATION OF A CONTROL BLOCK STRUCTURE TABLE MODULE

Explanation

The MQ dump formatting facility cannot be used to format control blocks. An error occurred during the startup process while attempting to LOAD one of the Control Block Structures Table modules (CSQWDST1, CSQWDST2, CSQWDST3, and CSQWDST4) from the MQ program library.

System action

Queue manager startup processing continues.

System programmer response

If you expect to experience problems, stop your queue manager, resolve the problem, and restart. If you do not anticipate that this error will cause problems, you can stop and restart the queue manager at a convenient time.

CSQW108E
UNABLE TO AUTOMATICALLY START '*type*' TRACE

Explanation

System parameters indicated that an MQ trace should be started automatically during queue manager initialization, but the queue manager was unable to start the trace.

System action

Queue manager initialization continues.

System programmer response

Start the trace with the START TRACE command after queue manager initialization is complete.

CSQW109E
TRACE INITIALIZATION PARAMETERS UNAVAILABLE, DEFAULTS USED FOR '*type*' TRACE

Explanation

The trace function was unable to access the trace initialization parameters defined by the CSQ6SYSP macro. Default values as defined by that macro are assumed for trace parameters.

System action

Queue manager initialization continues.

System programmer response

Determine if the system parameter load module (the default version is called CSQZPARM) is missing or inaccessible. Trace can be started with the START TRACE command.

CSQW120E
DEST VALUE IS INVALID FOR '*type*' TRACE

Explanation

A trace command has been entered, but the specified destination value is not valid for the trace type requested.

System action

Processing for the TRACE command is terminated.

System programmer response

If a START TRACE command was entered, specify a valid destination for the trace. Otherwise, a DISPLAY TRACE command can be issued to determine what traces are currently active. See MQSC commands for information about valid destinations.

CSQW121E
CLASS VALUE IS INVALID FOR '*type*' TRACE

Explanation

A trace command has been entered, but the specified class value is not valid for the trace type requested.

System action

Processing for the TRACE command is terminated.

System programmer response

If a START TRACE command was entered, specify a valid class for the trace. Otherwise, a DISPLAY TRACE command can be issued to determine what options are currently active. See MQSC commands for information about valid classes.

CSQW122E
'*keyword*' IS NOT VALID FOR '*type*' TRACE

Explanation

A trace command has been entered, but *keyword* is not valid for the trace type specified.

System action

Processing for the TRACE command is terminated.

System programmer response

Either the named keyword must be omitted from the command, or a different type of trace must be specified. See MQSC commands for information about valid combinations of keywords and trace types.

CSQW123I
csect-name TRACE RECORDING HAS BEEN RESUMED ON *dest*

Explanation

dest destination has resumed acceptance of trace data after an error.

System action

Data recording is resumed.

CSQW124E
csect-name '*type*' TRACE TERMINATED RC=*code* RMID=*nn*

Explanation

During processing *type* trace, processing ended due to an error. A trace type of blank indicates all tracing has stopped. RMID, displayed in decimal, identifies the resource manager. For information on IBM MQ RMIDs, see the TRACE commands in MQSC commands.

code, displayed in hexadecimal, specifies the return, reason, or abend code associated with the action. Refer to "IBM MQ for z/OS codes" on page 943 for information about these codes.

For codes caused by errors that are not generated by IBM MQ for z/OS, see the z/OS System completion codes for further information.

Further collection of the named trace is stopped. If it is necessary to resume collection of the trace, a START TRACE command can be issued. However if another error is experienced, the problem should be resolved before starting the trace collection again.

System action

Processing for the named trace type is stopped. The message is not externalized by the functional recovery routine, but is output whenever an IFC event is driven at a later time. A trace type of blank indicates all tracing has stopped.

System programmer response

Investigate the reasons for the error. If necessary to collect the named trace, issue a START TRACE command to resume processing.

CSQW125E
MULTIPLE VALUES NOT ALLOWED FOR *keyword* AND *keyword*

Explanation

Multiple values were specified for both of the named keywords. At most one of these keywords is allowed multiple values on a single command.

System action

Processing for the command is terminated.

System programmer response

Reenter a valid command. See MQSC commands for additional information.

CSQW126E
'*type*' TRACE NOT ALLOWED, ACTIVE TRACE TABLE FULL

Explanation

The *type* trace cannot be started because the active trace table has reached the maximum number of active traces allowed.

System action

Processing for the command is terminated.

System programmer response

Use the DISPLAY TRACE command to see if an active trace could be stopped. An active trace must be stopped before any other start trace command will be processed.

CSQW127I
CURRENT TRACE ACTIVITY IS -

Explanation

This message is issued in response to the DISPLAY TRACE command. For each trace that is active, the message indicates the trace number, the type of trace, the class(es) within type, the destination(s) for the trace entries, the user ID, and the RMID(s), as follows:

```
TNO TYPE CLASS DEST USERID RMID  tno type class dest userid rmid ⋮  END OF TRACE REPORT
```

The trace number *tno* can be:

01-03
A trace started automatically when the queue manager started, or a trace started by a START TRACE command.

04-32
A trace started by a START TRACE command.

00
The global trace started automatically when the channel initiator started.

Notes:

1. For TRACE(S) CLASS(4) (channel initiator statistics), the traces will only be gathered when the channel initiator is active and message CSQX128I has been output.
2. For TRACE(A) CLASS(4) (channel accounting), the traces will only be gathered when the channel initiator is active and message CSQX126I has been output.

CSQW130I
'*type*' TRACE STARTED, ASSIGNED TRACE NUMBER *tno*

Explanation

In response to a command, or automatically during queue manager initialization, a *type* trace has been started and assigned the trace number *tno*. Multiple messages are possible when the start command specifies multiple user identifiers.

System action

Processing for the request continues. If the specified trace applies to the channel initiator, a request will be queued: see message CSQW152I.

CSQW131I
STOP TRACE SUCCESSFUL FOR TRACE NUMBER(S) *tno*,...

Explanation

In response to a command, the trace number(s), *tno,...*, have been stopped. Up to five trace numbers can be listed. If more than five traces have been stopped, another CSQW131I message is sent.

System action

Processing for the request continues. If the specified trace applies to the channel initiator, a request will be queued: see message CSQW152I.

CSQW132I
ALTER TRACE SUCCESSFUL FOR TRACE NUMBER *tno*

Explanation

The trace number *tno* has been altered.

System action

Processing for the request continues.

CSQW133E
csect-name TRACE DATA LOST, *dest* NOT ACCESSIBLE RC=*code*

Explanation

The destination specified stopped accepting trace data during a trace. Some external condition caused the data rejection. The reason for the error is defined by the return code (RC). The value of *code* can be:

- The hexadecimal return code from SMF. See the *z/OS MVS System Management Facilities (SMF)* documentation for the specific value.
- The hexadecimal return code from the GTF request

04
GTF trace and/or USR tracing is not active

- The hexadecimal return code from the SRV request

10
The serviceability routine is absent

xx
The serviceability routine return code

System action

Trace processing continues, although data is lost.

System programmer response

Investigate the GTF or SMF facility to determine why data is not being accepted. You can issue a START TRACE command to record the data at another destination. The DISPLAY TRACE command shows what types of data were recorded at the specified destination.

See the *z/OS MVS System Management Facilities (SMF)* documentation for an explanation of the return code value.

CSQW135I
'*type*' TRACE ALREADY ACTIVE, TRACE NUMBER *tno*

Explanation

type trace was already active with trace number *tno*.

System action

Processing for the trace already in progress will continue.

CSQW137I
SPECIFIED TRACE NOT ACTIVE

Explanation

Either:

- A command requested action for a specific trace, but that trace could not be found in the active trace table.
- A command requested action for all traces, but there are no traces active.

System action

Processing for the command continues.

System programmer response

Issue an unqualified DISPLAY TRACE command (that is, DISPLAY TRACE(*) without any other keywords) to determine all the active trace entries.

CSQW138E
IFCID *ifcid-number* IS INVALID

Explanation

The specified IFCID number is outside the range of valid IFCID numbers or is an IFCID number which is not allowed on a trace command.

System action

Processing of the trace command is terminated before any trace functions are performed.

System programmer response

See the TRACE commands in MQSC commands and Line trace for more information.

CSQW144E
CHANNEL INITIATOR NOT ACTIVE

Explanation

TRACE(CHINIT) was specified, but the channel initiator is not active.

System action

The command is not actioned.

System programmer response

Issue the START CHINIT command to start the channel initiator, and reissue the command.

CSQW149E
RMID 231 IS OBSOLETE - USE TRACE(CHINIT)

Explanation

The command specifies RMID 231, which was formerly used for channel initiator traces, but is now obsolete. For channel initiator traces, specify TRACE(CHINIT).

System action

The command is not actioned.

System programmer response

Issue the command correctly. If both queue manager and channel initiator tracing is required, issue two separate commands.

CSQW152I
TRACE REQUEST FOR CHANNEL INITIATOR QUEUED

Explanation

Initial processing for a trace command has completed successfully. The command requires further action by the channel initiator, for which a request has been queued.

System action

A request has been queued for the channel initiator. Further messages will be produced when the command has been completed.

CSQW153E
csect-name STORAGE NOT AVAILABLE FOR NEW TRACE TABLE

Explanation

There is insufficient storage in ECSA for a new global trace table as requested by a previous SET SYSTEM TRACTBL command.

System action

Processing continues using the existing global trace table.

System programmer response

Investigate how ECSA storage is being used. Issue a further SET SYSTEM TRACTBL command to set the trace table size to an acceptable value.

CSQW200E
Error during STORAGE OBTAIN macro. Return code= *rc*

Explanation

The z/OS STORAGE macro was issued to obtain storage for the trace formatter. The request failed with return code *rc*.

System action

Formatting of control blocks stops, and a hexadecimal dump of the record is produced. (This might be only part of the logical record.)

System programmer response

See the appropriate volume of the *z/OS MVS Assembler Services Reference* manual for information about the STORAGE *rc*. You can usually resolve this problem by increasing the size of your TSO or batch region. When the problem has been solved, retry the operation.

CSQW201E
Error during STORAGE RELEASE macro. Return code= *rc*

Explanation

The z/OS STORAGE macro was issued to release some storage. The request failed with return code *rc*.

System action

Formatting of control blocks stops, and a hexadecimal dump of the record is produced. (This might be only part of the logical record.)

System programmer response

Try processing the dump again. If the problem persists, note the value of *rc*, and contact IBM Support.

CSQW202E
Incomplete trace record detected

Explanation

A long trace record has been segmented, and the start record for the record currently being processed has not been processed.

This usually occurs when records within a time range have been selected for processing. The record with the start of segment flag is probably before the start of the selected time interval. This can also occur if the Generalized Trace Facility (GTF) is unable to write all records to the GTF data set.

System action

A hexadecimal dump of the record is produced, and formatting continues with the next record. (You will receive this message for each subsequent part of this logical record.)

System programmer response

Select a slightly earlier start time for your time interval (one tenth of a second for example) and retry the operation. If this is not successful, it is possible that your trace table has wrapped, and the start record has been overwritten.

CSQW204E
Internal error

Explanation

An internal error has occurred.

System action

A hexadecimal dump of the record is produced, and formatting continues with the next record. This message might be followed by message CSQW202E.

System programmer response

Try processing the dump again. If the problem persists, contact IBM Support.

CSQW205E
Internal error

Explanation

An internal error has occurred.

System action

This, and all subsequent records are displayed in hexadecimal. IBM MQ trace formatting is suppressed.

System programmer response

Try processing the dump again. If the problem persists, contact IBM Support.

CSQW206I
Accounting record

Explanation

This message identifies this record as an accounting record.

System action

A hexadecimal dump of the record is produced, and formatting continues with the next record.

CSQW207I
A Null Self Defining section was detected

Explanation

The MQ trace formatter has detected a self-defining section of zero length.

System action

Formatting continues with the next self-defining section.

CSQW208E
Invalid address detected

Explanation

The MQ trace formatter has been passed an invalid address. The address is in low storage.

System action

Formatting of the record is suppressed. Formatting continues with the next record.

CSQW209I
A null length data item was detected

Explanation

The MQ trace formatter detected a data item of zero length.

System action

Formatting continues with the next data item.

CSQW210E
Invalid record detected

Explanation

The format of a record was different from the format expected by the IBM MQ trace formatter.

System action

A hexadecimal dump is produced, and formatting continues with the next record.

System programmer response

Try processing the dump again. If the problem persists, contact your z/OS support center.

CSQW701E
csect-name ENFREQ request failed, RC=*rc*

Explanation

A z/OS ENFREQ request failed. *rc* is the return code (in hexadecimal) from the request.

System action

Processing continues.

System programmer response

See the appropriate volume of the *z/OS MVS Authorized Assembler Services Reference* documentation for information about the return codes from the ENFREQ command.

z/OS *Distributed queuing messages (CSQX...)*

CSQX000I

IBM MQ for z/OS V*n*

Severity
0

Explanation

This message is issued when the channel initiator starts, and shows the release level.

CSQX001I

csect-name Channel initiator starting

Severity
0

Explanation

The channel initiator address space is starting, in response to a START CHINIT command.

System action

Channel initiator startup processing begins. Message CSQX022I is sent when the startup process has completed.

CSQX002I

csect-name Queue sharing group is *qsg-name*

Severity
0

Explanation

This is issued during channel initiator startup processing or in response to the DISPLAY CHINIT command if the queue manager that the channel initiator uses is in a queue sharing group.

System action

Processing continues.

CSQX003I

csect-name Obsolete parameter module ignored

Severity
0

Explanation

The START CHINIT command specified a parameter module name using the PARM keyword. The use of a channel initiator parameter module is obsolete, so the name is ignored.

System action

Processing continues.

System programmer response

Channel initiator parameters are specified by queue manager attributes. Use the ALTER QMGR command to set the values you want.

CSQX004I
Channel initiator storage usage :

Local storage: used 31-*used* MB, free 31-*free* MB
Above bar: used 64-*used*, free 64-*free*

Explanation

Displays the amount of virtual storage currently used and available in the extended private region, and above the bar. Values are approximate.

This message is logged at channel initiator start and then either every hour if the usage does not change or when the memory usage changes (up or down) by more than 2%.

The amount of currently used extended private region storage is also given in the *qcctstus* field in the SMF 115, subtype 231, record. 64-bit storage can be obtained from RMF, and is given in the *qcctstab* field in the SMF 115, subtype 231, record.

System action

Processing continues.

System programmer response

No action is required at this time. However, a frequent occurrence of this message might be an indication that the system is operating beyond the optimum region for the current configuration.

CSQX005E

csect-name Channel initiator failed to start

Severity
8

Explanation

A severe error, as reported in the preceding messages, occurred during channel initiator startup processing.

System action

The channel initiator started task ends.

System programmer response

Investigate the problem reported in the preceding messages.

CSQX006E

csect-name Channel initiator failed while stopping

Severity
8

Explanation

A severe error, as reported in the preceding messages, occurred during channel initiator termination processing.

System action

The channel initiator started task ends.

System programmer response

Investigate the problem reported in the preceding messages.

CSQX007E

csect-name Unable to connect to queue manager *qmgr-name*, MQCC=*mqcc* MQRC=*mqrc* (*mqrc-text*)

Severity
8

Explanation

An attempt by the channel initiator to connect to the queue manager was unsuccessful.

System action

If the error occurred during the channel initiator startup procedure, the channel initiator does not start. In other cases, the component where the error occurred (message channel agent, dispatcher, adapter subtask, SSL server subtask, repository manager, or listener) does not start and the function it provides is unavailable; in most cases, the end result is that the channel initiator ends.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form).

If you are unable to solve the problem, contact IBM Support.

CSQX008E

csect-name Unable to disconnect from queue manager *qmgr-name*, MQCC=*mqcc* MQRC=*mqrc* (*mqrc-text*)

Severity
4

Explanation

An attempt by the channel initiator to disconnect from the queue manager was unsuccessful.

System action

Processing continues.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form).

If you are unable to solve the problem, contact IBM Support.

CSQX009I

csect-name Channel initiator stopping

Severity
8

Explanation

A severe error, as reported in the preceding messages, occurred during channel initiator processing; the channel initiator is unable to continue.

System action

The channel initiator ends.

System programmer response

Investigate the problem reported in the preceding messages.

CSQX010I

csect-name Channel initiator stopped

Severity
0

Explanation

The channel initiator ended following an error, as reported in the preceding messages.

System action

None.

CSQX011I

csect-name Client attachment available

Severity
0

Explanation

Clients can be attached to and MQI channels can be used with the channel initiator.

System action

The channel initiator startup processing continues.

CSQX012E

csect-name Unable to open *ddname* data set

Severity
4

Explanation

The *ddname* data set could not be opened, as reported in the preceding messages.

System action

Processing continues, but functions that require the data set will be inhibited. For example, if the exit library data set CSQXLIB cannot be opened, user channel and channel auto-definition exits will not be available, and channels that use them will not start. If the error information data set CSQSNAP cannot be opened, the error information will be lost.

System programmer response

Investigate the problem reported in the preceding messages.

CSQX013I

csect-name Address conflict for listener, port *port* address *ip-address*, TRPTYPE=TCP INDISP=*disposition*

Severity
4

Explanation

A STOP LISTENER or START LISTENER command was issued specifying TRPTYPE(*trptype*) and INDISP(*disposition*), but that listener was already active for a port and IP address combination that conflicted with the requested port and IP address. If *ip-address* is '*', all IP addresses were requested.

The port and IP address combination specified must match a combination for which the listener is active. It cannot be a superset or a subset of that combination.

System action

None.

System programmer response

Reissue the command correctly if necessary.

CSQX014E

csect-name Listener exceeded channel limit, TRPTYPE=*trptype* INDISP=*disposition*

Severity
8

Explanation

The number of current channels using the indicated communications system *trptype* is the maximum allowed. The listener cannot accept an incoming request to start another channel; if the maximum is

0, the listener itself cannot start. (The name of the channel requested cannot be determined because the listener could not accept the request.) Current channels include stopped and retrying channels as well as active channels.

disposition shows which type of incoming requests the listener was handling:

QMGR
those directed to the target queue manager

GROUP
those directed to the queue sharing group.

The maximum allowed is specified in the TCPCHL or LU62CHL queue manager attribute, but may be reduced if a dispatcher fails, or if TCP/IP resources are restricted (as reported by message CSQX118I).

System action

The channel or listener does not start.

System programmer response

If the maximum allowed is zero, communications using the indicated system *trptype* are not allowed, and no such channels can be started. The listener also cannot be started. If the maximum allowed is non-zero, wait for some of the operating channels to end before restarting the remote channel, or use the ALTER QMGR command to increase TCPCHL or LU62CHL.

CSQX015I

csect-name started dispatchers started, *failed* failed

Severity
0

Explanation

The channel initiator startup procedure has started the requested number of dispatchers; *started* dispatchers started successfully and *failed* dispatchers did not start.

System action

The channel initiator startup processing continues. The number of current TCP/IP and LU 6.2 channels allowed will be reduced proportionately if some dispatchers did not start.

System programmer response

If the message indicates that some dispatchers failed, investigate the problem reported in the preceding messages.

CSQX016I

csect-name Listener already started, TRPTYPE=*trptype* INDISP=*disposition*

Severity
0

Explanation

A START LISTENER command was issued specifying TRPTYPE(*trptype*) and INDISP(*disposition*), but that listener was already active.

System action

None.

CSQX017I

csect-name Listener already started, port *port* address *ip-address*, TRPTYPE=TCP TRPTYPE=TCP INDISP=*disposition*

Severity
0

Explanation

A START LISTENER command was issued specifying TRPTYPE(TCP) and INDISP(*disposition*), but that listener was already active for the requested port and IP address. If *ip-address* is '*', all IP addresses were requested.

System action

None.

CSQX018I

csect-name Listener already stopped or stopping, TRPTYPE=*trptype* INDISP=*disposition*

Severity
0

Explanation

A STOP LISTENER or START LISTENER command was issued specifying TRPTYPE(*trptype*) and INDISP(*disposition*), but that listener was already stopped or in the process of stopping.

System action

None.

CSQX019I

csect-name Listener already stopped or stopping, port *port* address *ip-address*, TRPTYPE=TCP INDISP=*disposition*

Severity
0

Explanation

A STOP LISTENER or START LISTENER command was issued specifying TRPTYPE(*trptype*) and INDISP(*disposition*), but that listener was already stopped or in the process of stopping for the requested port and IP address. If *ip-address* is '*', all IP addresses were requested.

System action

None.

CSQX020I

csect-name Shared channel recovery completed

Severity
0

Explanation

The channel initiator startup procedure has successfully completed the shared channel recovery process, for channels that were owned by itself and for channels that were owned by other queue managers.

System action

Processing continues.

System programmer response

See message CSQM052I issued by the queue manager for more details.

CSQX021E

csect-name Shared channel recovery error

Severity
0

Explanation

The channel initiator startup procedure did not complete the shared channel recovery process, because an error occurred.

System action

The recovery process is ended; some channels might have been recovered, while others have not.

System programmer response

See the error messages (such as CSQM053E) issued by the queue manager for more details. When the problem has been resolved, either start any unrecovered channels manually, or restart the channel initiator.

CSQX022I

csect-name Channel initiator initialization complete

Severity
0

Explanation

Initialization of the channel initiator completed normally, and the channel initiator is ready for use. Note, however, that processing of the CSQINPX command data set might still be in progress; its completion is shown by message CSQU012I.

System action

None.

CSQX023I

csect-name Listener started, port *port* address *ip-address* TRPTYPE=*trptype* INDISP=*disposition*

Severity
0

Explanation

A listener has been started specifying TRPTYPE(*trptype*) and INDISP(*disposition*). This could either be because a START LISTENER command was issued, or because the listener was retrying. That listener is now active for the requested port and IP address. If *ip-address* is *, all IP addresses were requested.

System action

None.

CSQX024I

csect-name Listener stopped, port *port* address *ip-address* TRPTYPE=*trptype* INDISP=*disposition*

Severity
0

Explanation

A STOP LISTENER command was issued specifying TRPTYPE(*trptype*) and INDISP(*disposition*), or IBM MQ has tried to stop a listener because of a failure. That listener is no longer active for the requested port and IP address. If *ip-address* is *, all IP addresses were requested.

System action

None.

CSQX026E

csect-name Unable to locate the trace header, RC=12

Severity
8

Explanation

The trace formatting routine was unable to locate the trace control information in the trace data space in a dump of the channel initiator address space.

System action

Formatting ends.

System programmer response

The most likely cause is that the dump has not been produced correctly. Re-create the dump, and try again.

CSQX027E

csect-name Unable to get storage, RC=*return-code*

Severity
8

Explanation

An attempt to obtain some storage failed. *return-code* is the return code (in hexadecimal) from the z/OS STORAGE service.

System action

The component where the error occurred (message channel agent, dispatcher, adapter subtask, SSL server subtask, listener, repository manager, supervisor, or trace formatter) usually ends; in many cases, the end result will be that the channel initiator ends.

System programmer response

Go to the appropriate volume of the *z/OS MVS Programming: Assembler Services Reference* manual for information about the return code from the STORAGE request.

CSQX028E

csect-name Unable to free storage, RC=*return-code*

Severity
8

Explanation

An attempt to release some storage failed. *return-code* is the return code (in hexadecimal) from the z/OS STORAGE service.

System action

The component where the error occurred (message channel agent, dispatcher, adapter subtask, SSL server subtask, repository manager, or listener) usually ignores the error and continues processing.

System programmer response

Go to the appropriate volume of the *z/OS MVS Programming: Assembler Services Reference* manual for information about the return code from the STORAGE request.

CSQX029I

csect-name Queue manager *qmgr-name* stopping, MQCC=*mqcc* MQRC=*mqrc* (*mqrc-text*)

Severity
0

Explanation

In response to an MQ API call, the queue manager notified the channel initiator that it is stopping.

System action

The channel initiator ends.

System programmer response

Refer to “API completion and reason codes” on page 1151 for information about *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form).

CSQX030I

csect-name '*type*' trace started, assigned trace number *tno*

Explanation

During channel initiator initialization, a *type* trace has been started automatically and assigned the trace number *tno*.

System action

Processing continues.

CSQX031E

csect-name Initialization command handler ended abnormally, reason=*00sssuuu*

Severity

8

Explanation

The initialization command handler, which processes the CSQINPX command data set, is ending abnormally. *sss* is the system completion code, and *uuu* is the user completion code (both in hexadecimal).

System action

The initialization command handler ends abnormally, but the channel initiator continues.

System programmer response

If a system completion code is shown, see the appropriate *System codes* manual for information about the problem; the message will normally be preceded by other messages giving additional information.

The most likely cause is erroneous definition of the CSQINPX and CSQOUTX data sets. For information about the initialization command handler and these data sets, see Initialization commands. If you are unable to solve the problem, contact IBM Support.

CSQX032I

csect-name Initialization command handler ended

Severity

4

Explanation

The initialization command handler, which processes the CSQINPX command data set, was terminated before completing all the commands because the channel initiator is stopping, and so cannot process any more commands.

System action

The initialization command handler ends.

System programmer response

Refer to the CSQOUTX data set for information about the commands that were processed. If the channel initiator is not stopping because of a STOP command, refer to the preceding messages for information about the problem causing it to stop.

For information about the initialization command handler, see Initialization commands.

CSQX033E

csect-name Channel initiator stopping because of errors

Severity
8

Explanation

A severe error, as reported in the preceding messages, occurred during channel initiator processing; the channel initiator is unable to continue.

System action

The channel initiator terminates.

System programmer response

Investigate the problem reported in the preceding messages.

CSQX034I

csect-name Channel initiator stopping because queue manager is stopping

Severity
0

Explanation

The queue manager notified the channel initiator that it is stopping.

System action

The channel initiator terminates.

CSQX035I

csect-name Connection to queue manager *qmgr-name* stopping or broken, MQCC=*mqcc* MQRC=*mqrc* (*mqrc-text*)

Severity
0

Explanation

In response to an MQ API call, the channel initiator found that its connection to the queue manager was no longer available.

System action

The channel initiator terminates.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form).

CSQX036E

csect-name Unable to open *object-type*(*name*), MQCC=*mqcc* MQRC=*mqrc* (*mqrc-text*)

Severity
8

Explanation

An MQOPEN call for *name* was unsuccessful; *object-type* indicates whether *name* is a queue name, queue manager name, namelist name, channel name, topic name, or authentication information name. (The channel initiator can access channel definitions and authentication information as objects using the MQ API.)

System action

The component where the error occurred (message channel agent, dispatcher, adapter subtask, SSL server subtask, repository manager, listener, or supervisor) terminates. In the case of a message channel agent, the associated channel will be stopped.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form).

The most common cause of the problem will be that the channel and queue definitions are incorrect.

CSQX037E

csect-name Unable to get message from *name*, MQCC=*mqcc* MQRC=*mqrc* (*mqrc-text*)

Severity
8

Explanation

An MQGET call for queue *name* was unsuccessful.

System action

The component where the error occurred (message channel agent, dispatcher, adapter subtask, SSL server subtask, repository manager, listener, or supervisor) terminates. In the case of a message channel agent, the associated channel will be stopped.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form).

CSQX038E

csect-name Unable to put message to *name*, MQCC=*mqcc* MQRC=*mqrc* (*mqrc-text*)

Severity
8

Explanation

An MQPUT call for queue *name* was unsuccessful.

System action

The component where the error occurred (message channel agent, dispatcher, adapter subtask, SSL server subtask, repository manager, listener, or supervisor) terminates. In the case of a message channel agent, the associated channel will be stopped.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form).

CSQX039E

csect-name Unable to close *name*, MQCC=*mqcc* MQRC=*mqrc* (*mqrc-text*)

Severity
4

Explanation

An MQCLOSE call for *name* was unsuccessful; *name* can be a queue name, queue manager name, namelist name, channel name, or authentication information name. (The channel initiator can access channel definitions and authentication information as objects using the IBM MQ API.)

System action

Processing continues.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form).

CSQX040E

csect-name Unable to inquire attributes for *name*, MQCC=*mqcc* MQRC=*mqrc* (*mqrc-text*)

Severity
8

Explanation

An MQINQ call for *name* was unsuccessful; *name* may be a queue name, queue manager name, namelist name, channel name, or authentication information name. (The channel initiator can access channel definitions and authentication information as objects using the MQ API.)

System action

The component where the error occurred (message channel agent, dispatcher, adapter subtask, SSL server subtask, repository manager, listener, or supervisor) terminates. In the case of a message channel agent, the associated channel will be stopped.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form).

CSQX041E

csect-name Unable to set attributes for *name*, MQCC=*mqcc* MQRC=*mqrc* (*mqrc-text*)

Severity
8

Explanation

An MQSET call for queue *name* was unsuccessful.

System action

The component where the error occurred (message channel agent, dispatcher, adapter subtask, SSL server subtask, listener, or supervisor) terminates. In the case of a message channel agent, the associated channel will be stopped.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form).

CSQX042E

csect-name Unable to define *comp* to CTRACE, RC=*rc* reason=*reason*

Severity
8

Explanation

The CTRACE component definitions (for component *comp*) required by the channel initiator could not be defined. *rc* is the return code and *reason* is the reason code (both in hexadecimal) from the z/OS CTRACE service.

System action

The channel initiator does not start.

System programmer response

Go to the appropriate volume of the *z/OS MVS Programming: Authorized Assembler Services Reference* manual for information about the return code from the CTRACE request.

If you are unable to solve the problem, contact IBM Support.

CSQX043E

csect-name Unable to delete *comp* from CTRACE, RC=*rc* reason=*reason*

Severity
4

Explanation

The CTRACE component definitions (for component *comp*) used by the channel initiator could not be deleted. *rc* is the return code and *reason* is the reason code (both in hexadecimal) from the z/OS CTRACE service.

System action

Channel initiator termination processing continues.

System programmer response

Go to the appropriate volume of the *z/OS MVS Programming: Authorized Assembler Services Reference* manual for information about the return code from the CTRACE request.

If you are unable to solve the problem, contact IBM Support.

CSQX044E

csect-name Unable to initialize PC routines, RC=*rc* reason=*reason*

Severity
8

Explanation

The PC routines required by the channel initiator could not be defined. The reason code *reason* shows which z/OS service failed:

00E74007
LXRES failed

00E74008
ETCRE failed

00E74009
ETCON failed

rc is the return code (in hexadecimal) from the indicated z/OS service.

System action

The channel initiator does not start.

System programmer response

Go to the appropriate volume of the *z/OS MVS Programming: Authorized Assembler Services Reference* manual for information about the return codes from:

- LXRES
- ETCRE
- ETCON

If you are unable to solve the problem, contact IBM Support.

CSQX045E
csect-name Unable to load *module-name*, reason=*ssssrrrr*

Explanation

The channel initiator was unable to load a required module. *ssss* is the completion code and *rrrr* is the reason code (both in hexadecimal) from the z/OS LOAD service.

System action

The component where the error occurred (message channel agent, dispatcher, adapter subtask, SSL server subtask, repository manager, or listener) does not start and the function it provides is unavailable; in many cases, the end result is that the channel initiator terminates.

System programmer response

Check the console for messages indicating why the module was not loaded.

Go to the appropriate volume of the *z/OS MVS Programming: Authorized Assembler Services Reference* manual for information about the return code from the LOAD request.

Ensure that the module is in the required library, and that it is referenced correctly. The channel initiator attempts to load this module from the library data sets under the STEPLIB DD statement of its started task JCL procedure xxxxCHIN.

CSQX046E

csect-name Unable to initialize data conversion services, reason=*reason*

Severity
8

Explanation

The data conversion services required by the channel initiator could not be initialized. The reason code *reason* shows why:

00C10002
Unable to load modules

00C10003
Insufficient storage

other
Internal error

System action

The channel initiator does not start.

System programmer response

Check the console for messages indicating that a module was not loaded. Ensure that the module is in the required library, and that it is referenced correctly. The channel initiator attempts to load this module from the library data sets under the STEPLIB DD statement of its started task JCL procedure xxxxCHIN.

If you are unable to solve the problem, contact IBM Support.

CSQX047E

csect-name Unable to commit messages for *name*, MQCC=*mqcc* MQRC=*mqrc* (*mqrc-text*)

Severity
8

Explanation

An MQCMIT call involving messages for queue *name* was unsuccessful.

System action

The component where the error occurred (supervisor) terminates.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form).

CSQX048I

csect-name Unable to convert message for *name*, MQCC=*mqcc* MQRC=*mqrc* (*mqrc-text*)

Severity
0

Explanation

A message being put to an IMS bridge queue *name* required data conversion, but the conversion was not successful.

System action

The message is put without conversion, and processing continues.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form).

CSQX049E

csect-name Unable to retrieve token for name *name*, RC=*rc*

Severity

8

Explanation

A token in a name/token pair required by the channel initiator could not be retrieved. *rc* is the return code (in hexadecimal) from the z/OS IEANTRT service.

System action

The channel initiator does not start.

System programmer response

Go to the appropriate volume of the *z/OS MVS Programming: Authorized Assembler Services Reference* manual for information about the return code from the IEANTRT request.

If you are unable to solve the problem, contact IBM Support.

CSQX050E

csect-name Unable to create access list for queue manager, RC=*rc*

Severity

8

Explanation

The channel initiator could not create the necessary storage access list for the queue manager to use. *rc* is the return code (in hexadecimal) from the z/OS ALESERV service.

System action

The channel initiator does not start.

System programmer response

Go to the appropriate volume of the *z/OS MVS Programming: Authorized Assembler Services Reference* manual for information about the return code from the ALESERV request.

If you are unable to solve the problem, contact IBM Support.

CSQX051E

csect-name Unable to share storage with the queue manager, RC=*rc*

Severity

8

Explanation

A request by the channel initiator to allow the queue manager to share some storage failed. *rc* is the return code (in hexadecimal) from the z/OS IARVSERV service.

System action

The channel initiator does not start.

System programmer response

Go to the appropriate volume of the *z/OS MVS Programming: Assembler Services Reference* manual for information about the return code from the IARVSERV request.

If you are unable to solve the problem, contact IBM Support.

CSQX052E

csect-name Timer task attach failed, RC=*return-code*

Severity
8

Explanation

The repository manager task could not be attached. *return-code* is the return code (in hexadecimal) from the z/OS ATTACH service.

System action

The channel initiator terminates.

System programmer response

See the appropriate volume of the *z/OS MVS Programming: Authorized Assembler Services Reference* manual for information about the return code from the ATTACH request. If you are unable to solve the problem, contact IBM Support.

CSQX053E

csect-name Error information recorded in CSQSNAP data set

Severity
8

Explanation

An internal error has occurred. Information about the error is written to the data set identified by the CSQSNAP DD statement of the channel initiator started task JCL procedure, xxxxCHIN.

System action

Processing continues.

System programmer response

Collect the items listed in the Problem Determination section and contact IBM Support.

CSQX054E

csect-name Repository manager ended abnormally, reason=*sssuuu-reason*

Severity
8

Explanation

The repository manager is ending abnormally because an error that cannot be corrected has occurred. *sss* is the system completion code, *uuu* is the user completion code, and *reason* is the associated reason code (all in hexadecimal).

System action

The repository manager ends abnormally, and a dump is normally issued. The channel initiator will attempt to restart it.

System programmer response

User completion codes are generally the result of errors detected by the Language Environment; see the *z/OS Language Environment Runtime Messages* manual for information about these codes. Otherwise, contact IBM Support to report the problem.

CSQX055E

csect-name Repository manager attach failed, RC=*return-code*

Severity
8

Explanation

The repository manager task could not be attached. *return-code* is the return code (in hexadecimal) from the z/OS ATTACH service.

System action

The channel initiator terminates.

System programmer response

See the appropriate volume of the *z/OS MVS Programming: Authorized Assembler Services Reference* manual for information about the return code from the ATTACH request. If you are unable to solve the problem, contact IBM Support.

CSQX056E

csect-name Preinitialization services request failed, function code=*func*, RC=*rc*

Severity
8

Explanation

A preinitialization services (CEEPIPI) call failed. *func* is the function code used (in decimal) and *rc* is the return code (in hexadecimal) from the call.

System action

The component where the error occurred (message channel agent or SSL server subtask) terminates. In the case of a message channel agent, the associated channel will be stopped.

System programmer response

See *z/OS Language EnvironmentVendor Interfaces* for information about the return code from the CEEPIPI call. If you are unable to solve the problem, contact IBM Support.

CSQX057E

csect-name Cluster cache task attach failed, RC=*return-code*

Severity
8

Explanation

The channel initiator cluster cache task could not be attached. *return-code* is the return code (in hexadecimal) from the z/OS ATTACH service.

System action

The channel initiator terminates.

System programmer response

See the appropriate volume of the *z/OS MVS Programming: Authorized Assembler Services Reference* manual for information about the return code from the ATTACH request. If you are unable to solve the problem, contact your IBM support center.

CSQX058E

csect-name Pause service *service-name* failed, RC=*return-code*

Severity
8

Explanation

An error occurred processing a pause element. *return-code* is the return code (in hexadecimal) from the z/OS pause service *service-name*.

System action

The component where the error occurred (message channel agent, repository manager, cluster cache extension task,) usually terminates; in many cases, the end result will be that the channel initiator terminates. This can also be issued without a subsequent abend and therefore without termination, as the error has been tolerated.

System programmer response

See the appropriate volume of the *z/OS MVS Programming: Authorized Assembler Services Reference* manual for information about the return code from the IEAVPSE pause service request. If you are unable to solve the problem, contact your IBM support center. No action is required when CSQX058E is issued during channel initiator startup, if the channel initiator had previously terminated abnormally.

CSQX059E

csect-name Unable to increase cluster cache

Severity
8

Explanation

The dynamic cluster cache cannot be increased because the channel initiator cluster cache task encountered an error.

System action

The channel initiator probably terminates.

System programmer response

Investigate the problem reported in any preceding messages.

CSQX060E

csect-name Queued Pub/Sub task attach failed, RC=*reason-code*

Severity
8

Explanation

The queued Publish/Subscribe task could not be attached. The *return-code* is the return code (in hexadecimal) from the z/OS ATTACH service.

System action

The channel initiator terminates.

System programmer response

See the appropriate volume of the *z/OS MVS Programming: Authorized Assembler Services Reference* manual for information about the return code from the ATTACH request. If you are unable to solve the problem, contact your IBM support center.

CSQX061E

csect-name Distributed Pub/Sub Offloader task attach failed, RC=*return-code*

Severity
8

Explanation

The Distributed Pub/Sub Offloader task could not be attached. *Return-code* is the return code (in hexadecimal) from the z/OS ATTACH service.

System action

The channel initiator terminates.

System programmer response

See the appropriate volume of the *z/OS MVS Programming: Authorized Assembler Services Reference* manual for information about the return code from the ATTACH request. If you are unable to solve the problem, contact IBM Support.

CSQX062E

csect-name Distributed Pub/Sub tasks have insufficient command authority

Severity
8

Explanation

The PSMODE queue manager attribute has a value other than DISABLED but the channel initiator has insufficient authority to issue the DISPLAY PUBSUB command. Until such authority is granted, distributed publish/subscribe is unavailable.

System action

The channel initiator attempts to restart the distributed Pub/Sub tasks at 1 minute intervals. This message is issued on each subsequent attempt until the required authority has been granted or publish/subscribe is disabled.

System programmer response

Grant the channel initiator the required authority to access the command server queues and issue the DISPLAY PUBSUB command. For the required security definitions, see Security considerations for the channel initiator on z/OS . Alternatively, if no publish subscribe operation is required, setting the PSMODE queue manager attribute to DISABLED prevents this message from being issued.

CSQX063I

csect-name Distributed Pub/Sub Offloader started

Severity
0

Explanation

The Distributed Pub/Sub Offloader task has started successfully.

System programmer response

None

CSQX064I

csect-name Distributed Pub/Sub Offloader stopped

Severity
0

Explanation

The Distributed Pub/Sub command Offloader task has stopped. This can be for one of three reasons:

- The channel initiator is stopping.

- The channel initiator is starting and the queues used by the distributed pub/sub offloader have not been defined because distributed pub/sub command processing is not required.
- An error has occurred.

System action

Processing continues, but distributed pub/sub is not available.

System programmer response

If an error has occurred, investigate the problem reported in the preceding messages.

CSQX065E

csect-name Unexpected error in distributed pub/sub Offloader

Severity
8

Explanation

The Distributed Pub/Sub command Offloader encountered an unexpected error

System action

Distributed publish/subscribe might no longer be available.

System programmer response

Investigate the problem reported in the preceding messages. If there are none or this does not resolve the problem contact IBM Support.

CSQX066E

csect-name Refresh proxy subscriptions failed

Severity
8

Explanation

A REFRESH QMGR TYPE(PROXYSUB) was issued, but could not complete. This could be because the Channel Initiator is shutting down, or as a result of an error.

System action

Processing continues, but remote subscriptions are not resynchronized.

System programmer response

If an error has occurred, investigate the problem reported in the preceding messages.

CSQX067E

csect-name Error removing non durable remote subscriptions

Severity
8

Explanation

The Pub/Sub Offloader task is ending but was unable to remove one or more remote proxy subscriptions. If no previous error has occurred, this is likely to have been triggered by Queue Manager shut down.

System action

Processing continues, but remote subscriptions might continue to exist which are no longer valid. This could cause a build-up of publications for this Queue Manager on remote transmission queues.

System programmer response

If the Queue Manager is to be restarted immediately, these subscriptions will be cleaned up when initial resynchronization with the cluster occurs. If this is not the case, proxy subscriptions might need to be manually removed using DELETE SUB on other Queue Managers in the cluster. Investigate the problem reported in the preceding messages to see why resynchronization failed.

CSQX068I

csect-name Channel initiator has scavenged *mm* MB of transmission buffers

Explanation

Displays the amount of virtual storage that has been freed by the channel initiator transmission buffer scavenger task. This virtual storage value is displayed in megabytes (1048576 bytes), and is an approximation.

This message is logged when the amount of virtual storage used by the channel initiator is more than 75%. If storage has been freed the CSQX004I message is issued.

System action

Processing continues.

System programmer response

No action is required at this time. However, a frequent occurrence of this message might indicate the system is operating beyond the optimum region for the current configuration.

CSQX069E

csect-name Distributed Pub/Sub Offloader ended abnormally, reason=*sssuuu-reason*

Severity

8

Explanation

The Distributed Pub/Sub Offloader task is ending abnormally because an error that cannot be corrected has occurred. *sss* is the system completion code, *uuu* is the user completion code, and *reason* is the associated reason code (all in hexadecimal).

System action

The Distributed Pub/Sub Offloader task ends abnormally, and a dump is normally issued. Distributed publish/subscribe is no longer available.

System programmer response

User completion codes are generally the result of errors detected by the Language Environment; see the *z/OS Language Environment Programming Guide* for information about these codes. Otherwise, contact your IBM support center to report the problem.

CSQX070I

csect-name CHINIT parameters ...

Severity

0

Explanation

The channel initiator is being started with the parameter values shown in the following messages: CSQX071I, CSQX072I, CSQX073I, CSQX074I, CSQX075I, CSQX076I, CSQX078I, CSQX079I, CSQX080I, CSQX081I, CSQX082I, CSQX085I, CSQX090I, CSQX091I, CSQX092I, CSQX094I, CSQX099I.

System action

The channel initiator startup processing continues.

System programmer response

Channel initiator parameters are specified by queue manager attributes. Use the ALTER QMGR command to set the values you want.

CSQX093I

csect-name WLM/DNS is no longer supported

Severity
4

Explanation

The QMGR attribute DNSWLM is set to YES. This feature is no longer supported by z/OS Communications Server.

System action

Processing continues, but registration to the WLM/DNS server will not be attempted.

System programmer response

Issue the command

```
ALTER QMGR DNSWLM(NO)
```

and consider using Sysplex Distributor instead. See Establishing a TCP connection Using Sysplex Distributor.

CSQX100E

csect-name Dispatcher failed to start, TCB=*tcb-name*

Severity
8

Explanation

A severe error, as reported in the preceding messages, occurred during dispatcher startup processing.

System action

The channel initiator will attempt to restart the dispatcher. The number of current TCP/IP and LU 6.2 channels allowed will be reduced proportionately.

System programmer response

Investigate the problem reported in the preceding messages.

CSQX101E

csect-name Dispatcher unable to schedule essential process *process*

Severity
8

Explanation

During dispatcher startup processing, one of the essential dispatcher processes (named *process*) could not be scheduled.

System action

The dispatcher does not start.

System programmer response

The most likely cause is insufficient storage. If increasing the available storage does not solve the problem, contact IBM Support.

CSQX102E

csect-name Dispatcher linkage stack error, TCB=*tcb-name*

Severity
8

Explanation

The dispatcher using TCB *tcb-name* found an inconsistency in the linkage stack.

System action

The dispatcher ends abnormally with completion code X'5C6' and reason code X'00E7010E', and a dump is issued. The channel initiator will attempt to restart it.

System programmer response

The most likely cause is incorrect use of the linkage stack by a user channel exit; exits must issue any MQ API calls and return to the caller at the same linkage stack level as they were entered. If exits are not being used, or if they do not use the linkage stack, contact IBM Support to report the problem.

CSQX103E

csect-name Dispatcher unexpected error, TCB=*tcb-name* RC=*return-code*

Severity
8

Explanation

The dispatcher using TCB *tcb-name* had an internal error.

System action

The dispatcher ends abnormally with completion code X'5C6' and reason code X'00E7010F', and a dump is issued. The channel initiator will attempt to restart it.

System programmer response

Contact IBM Support to report the problem.

CSQX104E

csect-name Unable to establish ESTAE, RC=*return-code*

Severity
8

Explanation

During startup processing, the recovery environment could not be set up. *return-code* is the return code (in hexadecimal) from the z/OS ESTAE service.

System action

The component that was starting (dispatcher, adapter subtask, SSL server subtask, supervisor, repository manager, or channel initiator itself) does not start.

System programmer response

See the *Description* of ESTAE in the appropriate volume of the *z/OS MVS Programming: Assembler Services Reference* manual for information about the return code from the ESTAE request. If you are unable to solve the problem, contact IBM Support.

CSQX106E

csect-name Unable to connect to TCP/IP using USS, service '*serv*' RC=*return-code* reason=*reason*

Severity
4

Explanation

Use of TCP/IP with the z/OS UNIX System Services (z/OS UNIX) sockets interface was requested, but an error occurred. *return-code* and *reason* are the return and reason codes (both in hexadecimal) from the z/OS UNIX service *serv* that gave the error.

The most likely causes are:

- The user ID that the channel initiator uses is not set up correctly for use with z/OS UNIX. For example, it may not have a valid OMVS segment defined or its security profile may be incomplete.
- The TCPNAME queue manager attribute does not specify a valid TCP/IP stack name. These stack names are defined in the SUBFILESYSTYPE NAME parameter in member BPXPRMxx for SYS1.PARMLIB.
- The MAXFILEPROC or MAXPROCUSER parameter in member BPXPRMxx for SYS1.PARMLIB is too small.

System action

Processing continues, but communications using TCP/IP with the z/OS UNIX sockets interface will not be available.

System programmer response

See the *z/OS UNIX System Services Messages and Codes* manual for information about the codes from the service request.

CSQX110E

csect-name User data conversion exit error, TCB=*tcb-name* reason=*sssuuu-reason*

Severity
8

Explanation

A process for the dispatcher using TCB *tcb-name* is ending abnormally because an error that cannot be corrected has occurred in a user data conversion exit. *sss* is the system completion code, *uuu* is the user completion code, and *reason* is the associated reason code (all in hexadecimal).

System action

The process ends abnormally, and a dump is normally issued. The channel is stopped, and must be restarted manually.

System programmer response

User completion codes are generally the result of errors detected by the Language Environment; see the *Language Environment for z/OS Runtime Messages* manual for information about these codes. If a system completion code is shown, see the *z/OS MVS System Codes* manual for information about the problem in your exit.

CSQX111E

csect-name User channel exit error, TCB=*tcb-name* reason=*sssuuu-reason*

Severity
8

Explanation

A process for the dispatcher using TCB *tcb-name* is ending abnormally because an error that cannot be corrected has occurred in a user channel exit. *sss* is the system completion code, *uuu* is the user completion code, and *reason* is the associated reason code (all in hexadecimal).

System action

The process ends abnormally, and a dump is normally issued. The channel is stopped, and must be restarted manually. For auto-defined channels, the channel does not start.

System programmer response

User completion codes are generally the result of errors detected by the Language Environment; see the *Language Environment for z/OS Runtime Messages* manual for information about these codes. If a system completion code is shown, see the *z/OS MVS System Codes* manual for information about the problem in your exit.

CSQX112E

csect-name Dispatcher process error, TCB=*tcb-name* reason=*sssuuu-reason*

Severity
8

Explanation

A process run by the dispatcher using TCB *tcb-name* is ending abnormally because an error that cannot be corrected has occurred. *sss* is the system completion code, *uuu* is the user completion code, and *reason* is the associated reason code (all in hexadecimal).

System action

The process ends abnormally, and a dump is normally issued. If the process is a message channel agent, the channel is stopped, and will need to be restarted manually.

System programmer response

User completion codes are generally the result of errors detected by the Language Environment; see the *Language Environment for z/OS Runtime Messages* manual for information about these codes. If a system completion code is shown, and you are using user channel exits, check that your exit is setting its parameter lists correctly; otherwise, contact your IBM support center.

CSQX113E

csect-name Dispatcher ended abnormally, TCB=*tcb-name* reason=*sssuuu-reason*

Severity
8

Explanation

The dispatcher using TCB *tcb-name* is ending abnormally because an error that cannot be corrected has occurred. *sss* is the system completion code, *uuu* is the user completion code, and *reason* is the associated reason code (all in hexadecimal).

System action

The dispatcher ends abnormally, and a dump is normally issued. The channel initiator terminates.

System programmer response

User completion codes are generally the result of errors detected by the Language Environment; see the *Language Environment for z/OS Runtime Messages* manual for information about these codes. If a system completion code is shown, and you are using user channel exits, check that your exit is setting its parameter lists correctly; otherwise, contact your IBM support center.

CSQX114E

csect-name Dispatcher failed, reason=*reason*

Severity
8

Explanation

A dispatcher ended abnormally, as reported in the preceding messages, and could not be restarted. *reason* shows the type of failure:

0000000A
Startup error

0000000B
Linkage stack error

0000000D
Uncorrectable error

other
Completion code in the form 00*sss**uuu*, where *sss* is the system completion code and *uuu* is the user completion code (both in hexadecimal).

System action

The channel initiator will attempt to restart the dispatcher. The number of current TCP/IP and LU 6.2 channels allowed will be reduced proportionately.

System programmer response

Investigate the problem reported in the preceding messages.

CSQX115E

csect-name Dispatcher not restarted - too many failures

Severity
8

Explanation

A dispatcher failed; because it had already failed too many times, the channel initiator did not attempt to restart it.

System action

The dispatcher is not restarted. The number of current TCP/IP and LU 6.2 channels allowed is reduced proportionately, and other processing capacity might be reduced.

System programmer response

Investigate the problems causing the dispatcher failures.

CSQX116I

csect-name Dispatcher restarted, *number* dispatchers active

Severity
0

Explanation

A dispatcher failed, but was successfully restarted by the channel initiator. *number* dispatchers are now active.

System action

Processing continues. The number of current TCP/IP and LU 6.2 channels allowed will be increased proportionately.

CSQX117I

csect-name Outgoing shared channels are restricted from starting for TCP communication

Severity
0

Explanation

A CHISERVP() service parm flag has been set which restricts the ability for this queue manager from being able to start an outgoing shared TCP channel. For more details on this flag contact IBM Support. .

System action

Processing continues. This queue manager is unable to start outgoing shared TCP channels, and will not be selected during IBM MQ workload balanced start of a shared channel. This restriction persists until the flag is disabled and the channel initiator is restarted.

CSQX118I

csect-name TCP/IP channel limit reduced to *nn*

Severity

0

Explanation

This is issued during channel initiator startup processing and in response to the DISPLAY CHINIT command if the maximum number of current TCP/IP channels allowed is less than is specified in the TCPCHL queue manager attribute. This error can occur because:

- TCP/IP resources are restricted. The UNIX Systems Services MAXFILEPROC parameter (specified in the BPXPRMxx member of SYS1.PARMLIB) controls how many sockets each task is allowed: that is, how many channels each dispatcher is allowed
- Some dispatchers have failed and not been restarted; the number of current TCP/IP channels allowed is reduced proportionately

System programmer response

If TCP/IP resources are restricted, consider increasing either the UNIX Systems Services MAXFILEPROC parameter or the number of dispatchers if you need more current TCP/IP channels.

CSQX119I

csect-name LU 6.2 channel limit reduced to *nn*

Severity

0

Explanation

This is issued during channel initiator startup processing and in response to the DISPLAY CHINIT command if the maximum number of current LU 6.2 channels allowed is less than is specified in the LU62CHL queue manager attribute. This can occur because some dispatchers have failed and not been restarted; the number of current LU 6.2 channels allowed will be reduced proportionately.

CSQX120I

csect-name Shared channel recovery started for channels owned by this queue manager

Severity

0

Explanation

The channel initiator startup procedure is starting the shared channel recovery process, for channels that are owned by itself.

System action

Processing continues

System programmer response

See message CSQM052I issued by the queue manager for more details.

CSQX121I

csect-name Shared channel recovery started for channels owned by other queue managers in the same QSG

Severity

0

Explanation

The channel initiator startup procedure is starting the shared channel recovery process, for channels that are owned by other queue managers.

System action

Processing continues

System programmer response

See message CSQM052I issued by the queue manager for more details.

CSQX122E

csect-name Failed to process channel accounting, RC=*retcode*

Severity
8

Explanation

The channel initiator SMF task encountered an error processing channel accounting data. *retcode* contains the hexadecimal return code.

System action

Processing continues.

System programmer response

Contact IBM Support.

CSQX123E

csect-name Failed to process channel initiator statistics, RC=*retcode*

Severity
8

Explanation

The channel initiator SMF task encountered an error processing channel initiator statistics data. *retcode* contains the hexadecimal return code.

System action

Processing continues.

System programmer response

Contact IBM Support.

CSQX124E

csect-name SMF task ended abnormally, RC=*retcode*, reason=*reason*

Severity
8

Explanation

The channel initiator SMF task ended abnormally. Possible values for *reason* are:

C59592
The channel initiator failed to notify the SMF task to shutdown. *retcode* is the return code from the z/OS IEAVRLS service.

C59593
The SMF task encountered an error entering, or resuming from, the paused state. *retcode* is the return code from the z/OS IEAVPSE service.

C59594
During initialization of the SMF task an error occurred obtaining a pause element token (PET). *retcode* is the return code from the z/OS IEAVAPE service.

C59595
During initialization of the SMF task an error occurred obtaining storage.

System action

The channel initiator attempts to reattach the SMF task, unless the error occurred during:

- Channel initiator shutdown
- Obtaining storage (reason C59595)

System programmer response

For reason C59595, check MEMLIMIT for the channel initiator, or refer to the 256MB recommended limit.

For the other reasons, contact IBM Support.

CSQX126I

csect-name Channel accounting collection started

Severity
0

Explanation

The channel initiator has started collecting channel accounting data.

System action

Channel accounting data for channels with STATCHL (HIGH|MED|LOW) is collected and written to the System Management Facility (SMF).

CSQX127I

csect-name Channel accounting collection stopped

Severity
0

Explanation

The channel initiator has stopped collecting channel accounting data.

System action

Channel accounting data that has been collected for channels with STATCHL (HIGH|MED|LOW) is written to the System Management Facility (SMF).

CSQX128I

csect-name Channel initiator statistics collection started

Severity
0

Explanation

The channel initiator has started collecting channel initiator statistics data.

System action

Channel initiator statistics data is collected and written to the System Management Facility (SMF).

CSQX129I

csect-name Channel initiator statistics collection stopped

Severity
0

Explanation

The channel initiator has stopped collecting channel initiator statistics data.

System action

Channel initiator statistics data that has been collected is written to the System Management Facility (SMF).

CSQX130E

csect-name queue-name is defined on a non-recoverable CF structure

Severity
8

Explanation

The shared channel synchronization queue *queue-name* is defined on a Coupling Facility (CF) structure that does not support recovery. This means that if the structure fails, shared channels might report message sequence errors, and might also lose messages.

System action

Processing continues.

System programmer response

Alter the CFSTRUCT object for the CF structure, where the shared channel synchronization queue is defined to RECOVER(YES), or plan to move the shared channel synchronization queue to the CSQSYSAPPL structure, which should be defined with RECOVER(YES).

CSQX140E

csect-name Adapter failed to start

Severity
8

Explanation

A severe error, as reported in the preceding messages, occurred during adapter subtask startup processing.

System action

The channel initiator will attempt to restart the adapter subtask.

System programmer response

Investigate the problem reported in the preceding messages.

CSQX141I

csect-name started adapter subtasks started, *failed* failed

Severity
0

Explanation

The channel initiator startup procedure has started the requested number of adapter subtasks; *started* adapter subtasks started successfully and *failed* adapter subtasks did not start.

System action

The channel initiator startup processing continues.

System programmer response

If the message indicates that some adapter subtasks failed, investigate the problem reported in the preceding messages.

CSQX142E

csect-name Adapter subtask failed to start, TCB=*tcb-name*

Severity
8

Explanation

A severe error, as reported in the preceding messages, occurred during adapter subtask startup processing.

System action

The channel initiator will attempt to restart the adapter subtask.

System programmer response

Investigate the problem reported in the preceding messages.

CSQX143E

csect-name Adapter subtask ended abnormally, TCB=*tcb-name* reason=*sssuuu-reason*

Severity
8

Explanation

The adapter subtask using TCB *tcb-name* is ending abnormally because an error that cannot be corrected has occurred. *sss* is the system completion code, *uuu* is the user completion code, and *reason* is the associated reason code (all in hexadecimal).

System action

The adapter subtask ends abnormally, and a dump is normally issued. The channel initiator will attempt to restart it.

System programmer response

User completion codes are generally the result of errors detected by the Language Environment; see the *Language Environment for z/OS Runtime Messages* manual for information about these codes. If a system completion code is shown, and you are using user channel exits, check that your exit is setting its parameter lists correctly; otherwise, contact your IBM support center.

CSQX144E

csect-name Adapter subtask attach failed, RC=*return-code*

Severity
8

Explanation

An adapter subtask could not be attached. *return-code* is the return code (in hexadecimal) from the z/OS ATTACH service.

System action

The adapter subtask is not restarted.

System programmer response

See the appropriate volume of the *z/OS MVS Programming: Authorized Assembler Services Reference* manual for information about the return code from the ATTACH request. If you are unable to solve the problem, contact IBM Support.

CSQX145E

csect-name Adapter subtask not restarted - too many failures

Severity
8

Explanation

An adapter subtask failed; because it had already failed too many times, the channel initiator did not attempt to restart it.

System action

The adapter subtask is not restarted; processing capacity might therefore be reduced.

System programmer response

Investigate the problems causing the adapter subtask failures.

CSQX146I

csect-name Adapter subtask restarted, *active* subtasks active

Severity
0

Explanation

A adapter subtask failed, but was successfully restarted by the channel initiator. *active* adapter subtasks are now active.

System action

Processing continues.

CSQX150E

csect-name SSL server failed to start

Severity
8

Explanation

A severe error, as reported in the preceding messages, occurred during SSL server subtask startup processing.

System action

The channel initiator will attempt to restart the SSL server subtask.

System programmer response

Investigate the problem reported in the preceding messages.

CSQX151I

csect-name started SSL server subtasks started, *failed* failed

Severity
0

Explanation

The channel initiator startup procedure has started the requested number of SSL server subtasks; *started* SSL server subtasks started successfully and *failed* SSL server subtasks did not start.

System action

The channel initiator startup processing continues.

System programmer response

If the message indicates that some SSL server subtasks failed, investigate the problem reported in the preceding messages.

CSQX152E

csect-name SSL server subtask failed to start, TCB=*tcb-name*

Severity
8

Explanation

A severe error, as reported in the preceding messages, occurred during SSL server subtask startup processing.

System action

The channel initiator will attempt to restart the SSL server subtask.

System programmer response

Investigate the problem reported in the preceding messages.

CSQX153E

csect-name SSL server subtask ended abnormally, TCB=*tcb-name* reason=*sssuuu-reason*

Severity
8

Explanation

The SSL server subtask using TCB *tcb-name* is ending abnormally because an error that cannot be corrected has occurred. *sss* is the system completion code, *uuu* is the user completion code, and *reason* is the associated reason code (all in hexadecimal).

System action

The SSL server subtask ends abnormally, and a dump is normally issued. The channel initiator will attempt to restart it.

System programmer response

User completion codes are generally the result of errors detected by the Language Environment; see the *Language Environment for z/OS Runtime Messages* manual for information about these codes. If a system completion code is shown, and you are using user channel exits, check that your exit is setting its parameter lists correctly; otherwise, contact your IBM support center.

CSQX154E

csect-name SSL server subtask attach failed, RC=*return-code*

Severity
8

Explanation

An SSL server subtask could not be attached. *return-code* is the return code (in hexadecimal) from the z/OS ATTACH service.

System action

The SSL server subtask is not restarted.

System programmer response

See the appropriate volume of the *z/OS MVS Programming: Authorized Assembler Services Reference* manual for information about the return code from the ATTACH request. If you are unable to solve the problem, contact IBM Support.

CSQX155E

csect-name SSL server subtask not restarted - too many failures

Severity
8

Explanation

An SSL server subtask failed; because it had already failed too many times, the channel initiator did not attempt to restart it.

System action

The SSL server subtask is not restarted; processing capacity might therefore be reduced.

System programmer response

Investigate the problems causing the SSL server subtask failures.

CSQX156I

csect-name SSL server subtask restarted, *active* subtasks active

Severity
0

Explanation

A SSL server subtask failed, but was successfully restarted by the channel initiator. *active* SSL server subtasks are now active.

System action

Processing continues.

CSQX160E

csect-name SSL communications unavailable

Severity
4

Explanation

SSLKEYR is required when communicating with the service.

SSL communications are requested but an error, as reported in the preceding messages, occurred during channel initiator startup processing.

System action

Processing continues.

System programmer response

Investigate the problem reported in the preceding messages. If you do not want to use SSL communications, set the SSLTASKS queue manager attribute to 0.

CSQX161E

csect-name SSL key repository name not specified

Severity
4

Explanation

SSLKEYR is required when communicating with the service.

SSL communications are requested but no SSL key repository name (SSLKEYR) is specified; that is, the SSLTASKS queue manager attribute is non-zero, but the SSLKEYR queue manager attribute is blank.

System action

Processing continues, but communications using SSL will not be available.

System programmer response

Use the ALTER QMGR command to specify a name for the SSL key repository with the SSLKEYR attribute, and restart the channel initiator. If you do not want to use SSL communications, set the SSLTASKS queue manager attribute to 0.

CSQX162E

csect-name SSL CRL namelist is empty or wrong type

Severity
4

Explanation

SSL communications are requested but the SSL authentication namelist specified by the SSLCRLNL queue manager attribute is empty or not of type AUTHINFO.

System action

If this message is displayed during CHINIT startup, then MQ communications using SSL are not available.

If the message is displayed after a change to the existing MQ SSL configuration and issuing the REFRESH SECURITY TYPE(SSL) command, then the changed MQ SSL configuration is rejected and the current MQ SSL configuration remains in force. This is to prevent a set of valid and working MQ SSL definitions being inadvertently deactivated by an incorrect change.

Processing continues.

System programmer response

Correct the definitions of the namelist, and start the channel initiator again. If you do not want to use SSL communications, set the SSLTASKS queue manager attribute to 0.

CSQX163I

csect-name SSL CRL namelist has too many names - first *n* used

Severity
4

Explanation

The SSL authentication namelist specified by the SSLCRLNL queue manager attribute has more names than are supported. The number supported is *n*.

System action

Processing continues; the excess names are ignored.

System programmer response

Correct the definitions of the namelist.

CSQX164E

csect-name Unable to access SSL key repository

Severity
4

Explanation

The SSL key repository, with a name that is specified by the SSLKEYR queue manager attribute, could not be accessed.

The most likely causes are:

- The specified key repository does not exist.
- The channel initiator does not have permission to read the specified key repository.
- The channel initiator was unable to connect to the LDAP server specified in an authentication information object listed in the SSL CRL namelist.
- When using shared keyrings, the name is not prefixed with 'userid/'.

System action

Processing continues, but communications using SSL will not be available. Channels using SSL communications will not start.

System programmer response

Check that:

- the SSL key repository name is specified correctly; if using a shared keyring, it is prefixed with 'userid/'
- the keyring specified as the SSL key repository exists, and the channel initiator has permission to read it
- the LDAP name is specified correctly and that it is available.

For more information, refer to SSL function return code 202 in the *z/OS Cryptographic Services System SSL Programming* manual.

CSQX165I

csect-name SSL key repository refresh already in progress

Severity
0

Explanation

A REFRESH SECURITY TYPE(SSL) command was issued, but an SSL key repository refresh was already in progress.

System action

The command is ignored. The refresh currently in progress continues.

CSQX166E

csect-name AuthInfo *auth-info-name* has wrong type

Severity
4

Explanation

The SSL authentication namelist specified by the SSLCRLNL queue manager attribute contains the name of an authentication information object that has an AUTHTYPE of OCSP.

System action

Processing continues, but communications using SSL will not be available.

System programmer response

Correct the definitions supplied in the namelist so that only authentication information objects with AUTHTYPE of CRLLDAP are named, and restart the channel initiator. If you do not want to use SSL communications, set the SSLTASKS queue manager attribute to 0.

CSQX168E

csect-name Certificate label cache not generated. Unable to start channel *channel-name*, connection *conn-id*

Severity
8

Explanation

An error occurred during channel initiation where the required certificate label cache to establish a TLS connection was unable to be generated. This failure prevents the channel connection identified by *channel-name* from starting.

System action

The channel will not start.

System programmer response

Use the preceding messages on the z/OS console to investigate the issue.

To re-attempt building the cache you can issue the REFRESH SECURITY TYPE(SSL) command.

CSQX169I

csect-name Channel initiator has scavenged mm MB of above the bar transmission buffers

Explanation

Displays the amount of above the bar virtual storage that has been freed by the channel initiator buffer management task. This virtual storage value is displayed in megabytes (1048576 bytes), and is an approximation.

This message is logged when the channel initiator successfully frees above the bar transmission buffers in response to a storage shortage.

System action

Processing continues.

System programmer response

No action is required.

CSQX179I

csect-name Channel *channel-name* message reallocation is in progress, *msg-progress* messages of *msg-total* processed

Severity
0

Explanation

The channel *channel-name* is currently in message reallocation and the progression of this processing is *msg-progress* message processed out of *msg-total* number of messages total to be processed.

System action

The channel continues to reallocate messages. This process can take some time to complete if there are a large number of messages assigned to the channel on its transmission queue. An increase in CPU utilization might be observed during this time. Upon completion of the reallocation process the channel ends.

System programmer response

If reallocation is not required, for example because the destination queue manager is now available, reallocation can be interrupted using the command STOP CHANNEL MODE(FORCE).

CSQX180I

csect-name Channel *channel-name* completed message reallocation, *msg-processed* messages processed

Severity
0

Explanation

The channel *channel-name* has completed message reallocation processing, and processed *msg-processed* number of messages during this processing.

System action

The channel reallocation for this channel has finished and the channel ends.

System programmer response

Determine if messages have been successfully reallocated, and if the channel can be started again.

CSQX181E

csect-name Invalid response *response* set by exit *exit-name*

Severity
8

Explanation

The user exit *exit-name* returned an invalid response code (*response*, shown in hexadecimal) in the *ExitResponse* field of the channel exit parameters (MQCXP).

System action

Message CSQX190E is issued giving more details, and the channel stops. For auto-defined channels, the channel does not start.

System programmer response

Investigate why the user exit program set an invalid response code.

CSQX182E

csect-name Invalid secondary response *response* set by exit *exit-name*

Severity
8

Explanation

The user exit *exit-name* returned an invalid secondary response code (*response*, shown in hexadecimal) in the *ExitResponse2* field of the channel exit parameters (MQCXP).

System action

Message CSQX190E is issued giving more details, and the channel stops. For auto-defined channels, the channel does not start.

System programmer response

Investigate why the user exit program set an invalid secondary response code.

CSQX184E

csect-name Invalid exit buffer address *address* set by exit *exit-name*

Severity
8

Explanation

The user exit *exit-name* returned an invalid address for the exit buffer when the secondary response code in the *ExitResponse2* field of the channel exit parameters (MQCXP) is set to MQXR2_USE_EXIT_BUFFER.

System action

Message CSQX190E is issued giving more details, and the channel stops. For auto-defined channels, the channel does not start.

System programmer response

Investigate why the user exit program set an invalid exit buffer address. The most likely cause is failing to set a value, so that it is 0.

CSQX187E

csect-name Invalid header compression value set by exit *exit-name*

Severity
8

Explanation

The user exit *exit-name* returned a header compression value that was not one of those which were negotiated as acceptable when the channel started.

System action

Message CSQX190E is issued giving more details, and the channel stops. For auto-defined channels, the channel does not start.

System programmer response

Investigate why the user exit program set an invalid value. If necessary, alter the channel definitions so that the required compression value is acceptable.

CSQX188E

csect-name Invalid message compression value set by exit *exit-name*

Severity
8

Explanation

The user exit *exit-name* returned a message compression value that was not one of those which were negotiated as acceptable when the channel started.

System action

Message CSQX190E is issued giving more details, and the channel stops. For auto-defined channels, the channel does not start.

System programmer response

Investigate why the user exit program set an invalid value. If necessary, alter the channel definitions so that the required compression value is acceptable.

CSQX189E

csect-name Invalid data length *length* set by exit *exit-name*

Severity
8

Explanation

The user exit *exit-name* returned a data length value that was not greater than zero.

System action

Message CSQX190E is issued giving more details, and the channel stops. For auto-defined channels, the channel does not start.

System programmer response

Investigate why the user exit program set an invalid data length.

CSQX190E

csect-name Channel *channel-name* stopping because of error in exit *exit-name*, Id=*ExitId* reason=*ExitReason*

Severity
8

Explanation

The user exit *exit-name* invoked for channel *channel-name* returned invalid values, as reported in the preceding messages. *ExitId* shows the type of exit:

11
MQXT_CHANNEL_SEC_EXIT, security exit

12
MQXT_CHANNEL_MSG_EXIT, message exit

13
MQXT_CHANNEL_SEND_EXIT, send exit

14
MQXT_CHANNEL_RCV_EXIT, receive exit

15
MQXT_CHANNEL_MSG_RETRY_EXIT, message retry exit

16
MQXT_CHANNEL_AUTO_DEF_EXIT, auto-definition exit

and *ExitReason* shows the reason for invoking it:

11
MQXR_INIT, initialization

12
MQXR_TERM, termination

13
MQXR_MSG, process a message

14
MQXR_XMIT, process a transmission

15
MQXR_SEC_MSG, security message received

16
MQXR_INIT_SEC, initiate security exchange

17
MQXR_RETRY, retry a message

18
MQXR_AUTO_CLUSSDR, auto-definition of cluster-sender channel

28
MQXR_AUTO_CLUSRCVR, auto-definition of cluster-receiver channel

System action

The channel stops. The associated transmission queue may be set to GET(DISABLED) and triggering turned off. For auto-defined channels, the channel does not start.

System programmer response

Investigate why the user exit program set invalid values.

CSQX191I

csect-name Channel *channel-name* beginning message reallocation

Severity
0

Explanation

The channel *channel-name* is entering message reallocation because it cannot currently deliver messages to the destination queue manager.

System action

Messages that are not bound to a particular queue manager will be workload balanced. This may take some time if there are a large number of messages assigned to this channel. Check how many using the **DISPLAY CHSTATUS(*channel-name*) XQMSGSA** command.

System programmer response

If reallocation is not required, for example because the destination queue manager is now available, reallocation can be interrupted using **STOP CHANNEL MODE(FORCE)**.

CSQX192E

csect-name Channel *channel-name* unable to stop, message reallocation in progress

Severity
8

Explanation

A request to stop channel *channel-name* was made, but the channel cannot stop immediately because message reallocation is taking place.

System action

The channel continues to reallocate messages. This process can take some time to complete if there are a large number of messages assigned to the channel on its transmission queue. An increase in CPU utilization might be observed during this time. Upon completion of the reallocation process the channel ends.

System programmer response

The number of messages to be reallocated can be determined using the **DISPLAY CHSTATUS(*channel-name*) XQMSGSA** command.

Turn on the **MONCHL** attribute of the channel and check how many users are using the **DISPLAY CHSTATUS(*channel-name*) XQMSGSA** command. The value of **MONCHL** should be LOW,MEDIUM or HIGH. See MONCHL for further information.

If reallocation is not required, for example because the destination queue manager is not available, reallocation can be interrupted using the **STOP CHANNEL MODE(FORCE)** command.

CSQX196E

csect-name Data length *data-length* set by exit *exit-name* is larger than agent buffer length *ab-length*

Severity
8

Explanation

The user exit *exit-name* returned data in the supplied agent buffer, but the length specified is greater than the length of the buffer.

System action

Message CSQX190E is issued giving more details, and the channel stops. For auto-defined channels, the channel does not start.

System programmer response

Investigate why the user exit program set an invalid data length.

CSQX197E

csect-name Data length *data-length* set by exit *exit-name* is larger than exit buffer length *eb-length*

Severity
8

Explanation

The user exit *exit-name* returned data in the supplied exit buffer, but the length specified is greater than the length of the buffer.

System action

Message CSQX190E is issued giving more details, and the channel stops. For auto-defined channels, the channel does not start.

System programmer response

Investigate why the user exit program set an invalid data length.

CSQX199E

csect-name Unrecognized message code *ccc*

Severity
8

Explanation

An unexpected error message code has been issued by the channel initiator.

System action

Another upload attempt will be made at the next upload interval.

System programmer response

Use the error codes and explanation to identify the issue. Check the following:

- The `APIKey` and `ServiceURL` are specified in the `ReportingService` stanza in the CSQMINI DD card of the queue manager.
- The channel initiator has network access to the IBM Cloud® service.
- The channel initiator has a SSL keyring (SSLKEYR), and the IBM Cloud certificates are connected to the keyring.

CSQX201E

csect-name Unable to allocate conversation, channel *channel-name* connection *conn-id* TRPTYPE=*trptype* RC=*return-code* (*return-text*) reason=*reason*

Severity
8

Explanation

An attempt to allocate a conversation on connection *conn-id* was not successful. The associated channel is *channel-name*; in some cases its name cannot be determined and so is shown as '????'. *trptype* shows the communications system used:

TCP
TCP/IP

LU62
APPC/MVS

The return code from it was: (in hexadecimal) *return-code*, (in text) *return-text*. For some errors, there may also be an associated reason code *reason* (in hexadecimal) giving more information.

System action

The channel is not started.

System programmer response

The error may be due to an incorrect entry in the channel definition or some problems in the APPC setup. Correct the error and try again

It could also be that the listening program at the remote end is not running. If so, perform the necessary operations to start the listener for *trptype*, and try again.

See "Communications protocol return codes for z/OS" on page 1132 for information about the cause of the return code from the communications system. If using TCP/IP, see the *z/OS UNIX System Services Messages and Codes* manual for information about the reason code.

CSQX202E

csect-name Connection or remote listener unavailable, channel channel-name connection *conn-id*TRPTYPE=*trptype* RC=*return-code* (*return-text*) reason=*reason*

Severity

8

Explanation

An attempt to allocate a conversation was not successful because the connection *conn-id* was unavailable. The associated channel is *channel-name*; in some cases its name cannot be determined and so is shown as '????'. *trptype* shows the communications system used:

TCP
TCP/IP

LU62
APPC/MVS

The return code from it was: (in hexadecimal) *return-code*, (in text) *return-text*. For some errors, there might also be an associated reason code *reason* (in hexadecimal) giving more information.

System action

The attempt to start the channel is retried.

System programmer response

Try again later.

A likely cause is that the listener at the remote end was not running or has been started using the wrong port or LU name. If this is the case, perform the necessary operations to start the appropriate listener, and try again.

See "Communications protocol return codes for z/OS" on page 1132 for information about the cause of the return code from the communications system. If using TCP/IP, see the *z/OS UNIX System Services Messages and Codes* manual for information about the reason code.

If you receive reason code 468:

- You are not using the correct IP address.
- The listener for the port might not be active.
- A firewall does not allow the connection.

  When there are multiple links defined on a z/OS image, the image can have multiple host names depending on the link. You need to ensure that the correct host name is used as the sender end. Use the NETSTAT HOSTs command to display the host names on the image.

CSQX203E

csect-name Error in communications configuration, channel channel-name connection *conn-id* TRPTYPE=*trptype* RC=*return-code* (*return-text*) reason=*reason*

Severity

8

Explanation

An attempt to allocate a conversation on connection *conn-id* was not successful because of a communications configuration error. The associated channel is *channel-name*; in some cases its name cannot be determined and so is shown as '????'. *trptype* shows the communications system used:

TCP
TCP/IP

LU62
APPC/MVS

The return code from it was: (in hexadecimal) *return-code*, (in text) *return-text*. For some errors, there might also be an associated reason code *reason* (in hexadecimal) giving more information.

System action

The channel is not started.

System programmer response

See "Communications protocol return codes for z/OS" on page 1132 for information about the cause of the return code from the communications system.

Probable causes are:

- If the communications protocol is TCP/IP:
  - The connection name specified is incorrect, or that it cannot be resolved to a network address, or the name may not be in the name server. Correct the error and try again.
  - If the return code is zero, there is a name server problem. The OMVS command OPING usually fails in the same way. Resolve this failure and restart the channel. Check the `/etc/resolv.conf` file and check that the correct name server address is specified in the NSINTERADDR statement.
- If the communications protocol is LU 6.2:
  - One of the transmission parameters (MODENAME or TPNAME or PARTNER_LU) in the side information is incorrect, or that there is no side information for the symbolic destination name specified as the connection name. Correct the error and try again.
  - An LU 6.2 session has not been established, perhaps because the LU has not been enabled. Issue the z/OS command VARY ACTIVE if this is the case.

See the *z/OS UNIX System Services Messages and Codes* manual for information about the reason code.

CSQX204E

csect-name Connection attempt rejected, channel channel-name connection *conn-id* TRPTYPE=*trptype* RC=*return-code* (*return-text*) reason=*reason*

Severity
8

Explanation

An attempt to connect on connection *conn-id* was rejected. The associated channel is *channel-name*; in some cases its name cannot be determined and so is shown as '????'. *trptype* shows the communications system used:

TCP
TCP/IP

LU62
APPC/MVS

The return code from it was: (in hexadecimal) *return-code*, (in text) *return-text*. For some errors, there might also be an associated reason code *reason* (in hexadecimal) giving more information.

System action

The channel is not started.

System programmer response

Check the appropriate listener has been started on the remote end.

See "Communications protocol return codes for z/OS" on page 1132 for information about the cause of the return code from the communications system.

If the communications protocol is LU 6.2, it is possible that either the user ID or password supplied at the remote LU is incorrect. The remote host or LU may not be configured to allow connections from the local host or LU.

If the communications protocol is TCP/IP, it is possible that the remote host does not recognize the local host. See the *z/OS UNIX System Services Messages and Codes* manual for information about the reason code.

CSQX205E

csect-name Unable to resolve network address, channel channel-name connection *conn-id* TRPTYPE=TCP RC=*return-code* (*return-text*) reason=*reason*

Severity

8

Explanation

The supplied connection name *conn-id* could not be resolved into a TCP/IP network address. The associated channel is *channel-name*; in some cases its name cannot be determined and so is shown as '????'. *trptype* shows the communications system used:

TCP
TCP/IP

LU62
APPC/MVS

The return code from it was: (in hexadecimal) *return-code*, (in text) *return-text*. For some errors, there might also be an associated reason code *reason* (in hexadecimal) giving more information.

System action

The channel is not started.

System programmer response

Check the local TCP/IP configuration. Either the name server does not contain the host or LU name, or the name server was not available.

See "Communications protocol return codes for z/OS" on page 1132 for information about the cause of the return code from TCP/IP. See the *z/OS UNIX System Services Messages and Codes* manual for information about the reason code.

CSQX206E

csect-name Error sending data, channel *channel-name* connection *conn-id* (queue manager *qmgr-name*) TRPTYPE=*trptype* RC=*return-code* (*return-text*) reason=*reason*

Severity

8

Explanation

An error occurred sending data to *conn-id*, which might be due to a communications failure. The associated channel is *channel-name* and the associated remote queue manager is *qmgr-name*; in some cases the names cannot be determined and so are shown as '????'. *trptype* shows the communications system used:

TCP
TCP/IP

LU62
APPC/MVS

The return code from it was: (in hexadecimal) *return-code*, (in text) *return-text*. For some errors, there might also be an associated reason code *reason* (in hexadecimal) giving more information.

System action

The channel is stopped. The associated transmission queue might be set to GET(DISABLED) and triggering turned off.

System programmer response

See "Communications protocol return codes for z/OS" on page 1132 for information about the cause of the return code from the communications system. If using TCP/IP, see the *z/OS UNIX System Services Messages and Codes* manual for information about the reason code.

Note that the error might have occurred because the channel at the other end has stopped for some reason, for example an error in a receive user exit.

CSQX207E

csect-name Invalid data received, connection *conn-id* (queue manager *qmgr-name*) TRPTYPE=*trptype*

Severity
8

Explanation

Data received from connection *conn-id* was not in the required format. The associated remote queue manager is *qmgr-name*; in some cases its name cannot be determined and so is shown as '????'. The data that has been sent may come from something other than a queue manager or client. *trptype* shows the communications system used:

TCP
TCP/IP

LU62
APPC/MVS

System action

The data is ignored.

System programmer response

A likely cause is that an unknown host or LU is attempting to send data.

CSQX208E

csect-name Error receiving data, channel *channel-name* connection *conn-id* (queue manager *qmgr-name*) TRPTYPE=*trptype* RC=*return-code* (*return-text*) reason=*reason*

Severity
8

Explanation

An error occurred receiving data from connection *conn-id*, which may be due to a communications failure. The associated channel is *channel-name* and the associated remote queue manager is *qmgr-name*; in some cases the names cannot be determined and so are shown as '????'. *trptype* shows the communications system used:

TCP
TCP/IP

LU62
APPC/MVS

The return code from it was: (in hexadecimal) *return-code*, (in text) *return-text*. For some errors, there might also be an associated reason code *reason* (in hexadecimal) giving more information.

System action

The channel is stopped. The associated transmission queue may be set to GET(DISABLED) and triggering turned off.

System programmer response

See "Communications protocol return codes for z/OS" on page 1132 for information about the cause of the return code from the communications system. If using TCP/IP, see Return code 00000461 for more information about the reason code.

CSQX209E

csect-name Connection unexpectedly terminated, channel *channel-name* connection *conn-id* (queue manager *qmgr-name*) TRPTYPE=*trptype* RC=*return-code* (*return-text*)

Severity

8

Explanation

An error occurred receiving data from connection *conn-id*. The connection to the remote host or LU has unexpectedly terminated. The associated channel is *channel-name* and the associated remote queue manager is *qmgr-name*; in some cases the names cannot be determined and so are shown as '????'. *trptype* shows the communications system used:

TCP
TCP/IP

LU62
APPC/MVS

However, this message can also occur in cases where there is no error; for example, if the TCP/IP command TELNET is issued that is directed at the port which the channel initiator is using.

The return code from it was: (in hexadecimal) *return-code*, (in text) *return-text*. For some errors, there might also be an associated reason code *reason* (in hexadecimal) giving more information.

System action

If a channel is involved, it is stopped. The associated transmission queue may be set to GET(DISABLED) and triggering turned off.

System programmer response

Review the local and remote console logs for reports of network errors.

See"Communications protocol return codes for z/OS" on page 1132 for information about the cause of the return code from the communications system. If using TCP/IP, see the *z/OS UNIX System Services Messages and Codes* manual for information about the reason code.

CSQX210E

csect-name Unable to complete bind, channel *channel-name* connection *conn-id* TRPTYPE=LU62 RC=*return-code* (*return-text*) reason=*reason*

Severity

8

Explanation

An incoming attach request arrived on connection *conn-id*, but the local host or LU was unable to complete the bind. The associated channel is *channel-name*; in some cases its name cannot be determined and so is shown as '????'.

The return code from APPC/MVS allocate services was: (in hexadecimal) *return-code*, (in text) *return-text*. For some errors, there might also be an associated reason code *reason* (in hexadecimal) giving more information.

System action

The channel is not started.

System programmer response

Check the APPC/MVS configuration.

See "APPC/MVS return codes" on page 1136 for the cause of the return code from APPC/MVS allocate services, and the *z/OS MVS programming: Writing Servers for APPC/MVS* manual for more information.

CSQX212E

csect-name Unable to allocate socket, channel *channel-name* TRPTYPE=TCP RC=*return-code* (*return-text*) reason=*reason*

Severity
8

Explanation

A TCP/IP socket could not be created, possibly because of a storage problem. The associated channel is *channel-name*; in some cases its name cannot be determined and so is shown as '????'.

The return code from TCP/IP was: (in hexadecimal) *return-code*, (in text) *return-text*. For some errors, there might also be an associated reason code *reason* (in hexadecimal) giving more information.

System action

The channel is not started.

System programmer response

See "Communications protocol return codes for z/OS" on page 1132 for information about the cause of the return code from TCP/IP. See the *z/OS UNIX System Services Messages and Codes* manual for information about the reason code.

CSQX213E

csect-name Communications error, channel *channel-name* TRPTYPE=*trptype* function *func* RC=*return-code* (*return-text*) reason=*reason*

Severity
8

Explanation

An unexpected communications error occurred for a listener or a channel. If it was for a listener, the *csect-name* is CSQXCLMA, and the channel name is shown as '????'. If it was for a channel, the channel is *channel-name*; in some cases its name cannot be determined and so is shown as '????'.

trptype shows the communications system used:

TCP
TCP/IP

LU62
APPC/MVS

func is the name of the TCP/IP or APPC/MVS function that gave the error. In some cases the function name is not known and so is shown as '????'.

return-code is

- normally, the return code (in hexadecimal) from the communications system function
- for an LU 6.2 listener, it might be the reason code (in hexadecimal) from APPC/MVS allocate services

- if it is of the form 10009*nnn* or 20009*nnn*, it is a distributed queuing message code.

return-text is the text form of the return code.

For some errors, there might also be an associated reason code *reason* (in hexadecimal) giving more information.

System action

If the error occurred for a channel, the channel is stopped. For a listener, the channel is not started or, in some cases, the listener terminates.

System programmer response

See "Communications protocol return codes for z/OS" on page 1132 for information about the cause of the return code from the communications system.

For the ioctl function, if the return-code is `0000007A` and the reason is `12B20381`, the problem might be that the queue manager has been configured with TCPSTACK(MULTIPLE) and the TCPIP stack named in the TCPNAME parameter does not exist.

If this is the case, either start the specified TCPIP stack, or change the value of the TCPNAME parameter on the queue manager.

A distributed queuing message code *nnn* is generally associated with message CSQX*nnn*E, which will normally be issued previously. See that message explanation for more information. Where no such message is described, see "Distributed queuing message codes" on page 1147 for the corresponding message number.

Check for error messages on the partner system that might indicate the cause of the problem.

CSQX215E

csect-name Communications network not available, TRPTYPE=*trptype*

Severity
8

Explanation

An attempt was made to use the communications system, but it has not been started or has stopped. *trptype* shows the communications system used:

TCP
TCP/IP

LU62
APPC/MVS

System action

The channel or listener is not started.

System programmer response

Start the communications system, and try again.

CSQX218E

csect-name Listener not started - unable to bind, port *port* address *ip-address* TRPTYPE=TCP INDISP=*disposition* RC=*return-code*

Severity
8

Explanation

An attempt to bind the TCP/IP socket to the indicated listener port was not successful. *ip-address* is the IP address used, or '*' if the listener is using all IP addresses. The return code (in hexadecimal) from TCP/IP was *return-code*.

disposition shows which type of incoming requests the listener was handling:

QMGR
those directed to the target queue manager

GROUP
those directed to the queue sharing group.

System action

The listener is not started.

System programmer response

The failure could be due to another program using the same port number.

See “Communications protocol return codes for z/OS” on page 1132 for information about the return code from TCP/IP.

CSQX219E

csect-name Listener stopped - error creating new connection, TRPTYPE=TCP INDISP=*disposition*

Severity
8

Explanation

An attempt was made to create a new TCP/IP socket because an attach request was received, but an error occurred.

disposition shows which type of incoming requests the listener was handling:

QMGR
those directed to the target queue manager

GROUP
those directed to the queue sharing group.

System action

The listener stops. The channel initiator will attempt to restart it, at the intervals specified by the LSTRTMR queue manager attribute.

System programmer response

The failure might be transitory, try again later. If the problem persists, it might be necessary to stop some other jobs that use TCP/IP, or to restart TCP/IP.

CSQX220E

csect-name Communications network not available, channel *channel-name* TRPTYPE=*trptype*

Severity
8

Explanation

An attempt was made to use the communications system by a channel or a listener, but it has not been started or has stopped. If it was for a channel, the channel is *channel-name*; in some cases its name cannot be determined and so is shown as '????'. If it was for a listener, the channel name is again shown as '????'. *trptype* shows the communications system used:

TCP
TCP/IP

LU62
APPC/MVS

System action

The channel or listener is not started.

System programmer response

Start the communications system, and try again.

CSQX228E

csect-name Listener unable to start channel, channel *channel-name* TRPTYPE=*trptype* INDISP=*disposition* connection=*conn-id*

Severity
8

Explanation

An incoming attach request arrived from *conn-id*, but the listener for *trptype* could not start an instance of a channel to respond to it. The associated channel is *channel-name*; in some cases its name cannot be determined and so is shown as '????'.

disposition shows which type of incoming requests the listener was handling:

QMGR
those directed to the target queue manager

GROUP
those directed to the queue sharing group.

However, this message can also occur in cases where there is no error; for example, if the TCP/IP command TELNET is issued that is directed at the port which the channel initiator is using.

System action

If a channel is involved, it is not started.

System programmer response

The failure could be because the channel initiator is currently too busy; try again when there are fewer channels running. If the problem persists, increase the number of dispatchers used by the channel initiator.

CSQX234I

csect-name Listener stopped, TRPTYPE=*trptype* INDISP=*disposition*

Severity
0

Explanation

The specified listener terminated. This could be for a number of reasons including, but not limited to, those in the following list:

- a STOP command was issued
- the listener was retrying
- an error occurred in the communications system

trptype is the transport type.

disposition shows which type of incoming requests the listener was handling:

QMGR
those directed to the target queue manager

GROUP
those directed to the queue sharing group.

System action

Processing continues. If the listener was not deliberately stopped, the channel initiator will attempt to restart the listener, at the intervals specified by the LSTRTMR queue manager attribute.

System programmer response

If the listener was not deliberately stopped, look at any preceding messages relating to the channel initiator or to the TCP/IP, OMVS, or APPC address spaces to determine the cause.

CSQX235E

csect-name Invalid local address *local-addr*, channel *channel-name* TRPTYPE=*trptype* RC=*return-code* (*return-text*) reason=*reason*

Severity

8

Explanation

The supplied local address *local-addr* could not be resolved to a TCP/IP network address. The associated channel is *channel-name*; in some cases its name cannot be determined and so is shown as '????'. *trptype* shows the communications system used:

TCP

TCP/IP

LU62

APPC/MVS

The return code from it was: (in hexadecimal) *return-code*, (in text) *return-text*. For some errors, there might also be an associated reason code *reason* (in hexadecimal) giving more information.

System action

The channel is not started.

System programmer response

Check the local TCP/IP configuration. Either the name server does not contain the host name, or the name server was not available.

See “Communications protocol return codes for z/OS” on page 1132 for information about the cause of the return code from TCP/IP.

CSQX239E

csect-name Unable to determine local host name, channel *channel-name* TRPTYPE=TCP RC=*return-code* (*return-text*) reason=*reason*

Severity

8

Explanation

An attempt was made to start a channel or listener using TCP/IP, but the TCP/IP gethostname call failed. If it was for a channel, the channel is *channel-name*; in some cases its name cannot be determined and so is shown as '????'. If it was for a listener, the channel name is again shown as '????'.

The return code from it was: (in hexadecimal) *return-code*, (in text) *return-text*. For some errors, there might also be an associated reason code *reason* (in hexadecimal) giving more information.

System action

The channel or listener is not started.

System programmer response

See “Communications protocol return codes for z/OS” on page 1132 for information about the cause of the return code from TCP/IP.

CSQX250E

csect-name Listener ended abnormally, TRPTYPE=*trptype* INDISP=disposition, reason=*sssuuu-reason*

Severity

8

Explanation

The specified listener is ending abnormally because an error that cannot be corrected has occurred. *sss* is the system completion code, *uuu* is the user completion code, and *reason* is the associated reason code (all in hexadecimal).

disposition shows which type of incoming requests the listener was handling:

QMGR
those directed to the target queue manager

GROUP
those directed to the queue sharing group.

System action

The listener ends abnormally, and a dump is normally issued. The channel initiator will attempt to restart the listener, at the intervals specified by the LSTRTMR queue manager attribute.

System programmer response

User completion codes are generally the result of errors detected by the Language Environment; see the *z/OS Language Environment Programming Guide* for information about these codes. Otherwise, contact your IBM support center.

CSQX251I

csect-name Listener started, TRPTYPE=*trptype* INDISP=*disposition*

Severity
0

Explanation

The specified listener started successfully. This may be as a result of a START LISTENER command, or because the listener restarted automatically following an error.

disposition shows which type of incoming requests the listener was handling:

QMGR
those directed to the target queue manager

GROUP
those directed to the queue sharing group.

System action

Processing continues.

CSQX256E

csect-name Listener stopped - error selecting new connection, TRPTYPE=TCP INDISP=*disposition*

Severity
8

Explanation

An error occurred in the listener select processing. The listener was notified by TCP/IP, but no attach request was received.

disposition shows which type of incoming requests the listener was handling:

QMGR
those directed to the target queue manager

GROUP
those directed to the queue sharing group.

System action

The listener stops. The channel initiator will attempt to restart it, at the intervals specified by the LSTRTMR queue manager attribute.

System programmer response

The failure might be transitory, try again later. If the problem persists, it might be necessary to stop some other jobs that use TCP/IP, or to restart TCP/IP.

CSQX257I

csect-name Listener unable to create new connection, TRPTYPE=TCP INDISP=*disposition*

Severity
4

Explanation

An attempt was made to create a new TCP/IP socket because an attach request was received, but an error occurred.

disposition shows which type of incoming requests the listener was handling:

QMGR
those directed to the target queue manager

GROUP
those directed to the queue sharing group.

System action

The listener continues to run, but the connection is not created.

System programmer response

The failure might be transitory, try again later. If the problem persists, it might be necessary to stop some other jobs that use TCP/IP, or to restart TCP/IP.

CSQX258E

csect-name Listener stopped - error accepting new connection, TRPTYPE=TCP INDISP=*disposition*

Severity
8

Explanation

An error occurred in the listener accept processing. The listener was notified by TCP/IP, but no attach request was received.

disposition shows which type of incoming requests the listener was handling:

QMGR
those directed to the target queue manager

GROUP
those directed to the queue sharing group.

System action

The listener stops. The channel initiator will attempt to restart it, at the intervals specified by the LSTRTMR queue manager attribute.

System programmer response

The failure might be transitory, try again later. If the problem persists, it might be necessary to stop some other jobs that use TCP/IP, or to restart TCP/IP.

CSQX259E

csect-name Connection timed out, channel *channel-name* connection *conn-id* (queue manager *qmgr-name*) TRPTYPE=*trptype*

Severity
8

Explanation

The connection *conn-id* timed out. The associated channel is *channel-name* and the associated remote queue manager is *qmgr-name*; in some cases the names cannot be determined and so are shown as '????'. *trptype* shows the communications system used:

TCP
TCP/IP

LU62
APPC/MVS

Probable causes are:

- A communications failure.
- For a message channel, if the Receive Timeout function is being used (as set by the RCVTIME, RCVTTYPE, and RCVTMIN queue manager attributes) and no response was received from the partner within this time.
- For an MQI channel, if the Client Idle function is being used (as set by the DISCINT server-connection channel attribute) and the client application did not issue an MQI call within this time.

System action

The channel stops.

System programmer response

For a message channel, check the remote end to see why the time out occurred. Note that, if retry values are set, the remote end will restart automatically. If necessary, set the receive wait time for the queue manager to be higher.

For an MQI channel, check that the client application behavior is correct. If so, set the disconnect interval for the channel to be higher.

CSQX261E

csect-name No suitable IP stack available, channel *channel-name*, connection *conn-id*

Severity
8

Explanation

An attempt to allocate a conversation on connection *conn-id* for channel *channel-name* using TCP/IP communications was not successful because the IP stack used did not support the IP address family required for the connection.

System action

The channel is not started.

System programmer response

If the channel's CONNAME attribute resolves to an IPv6 address, then ensure the stack being used by the combination of the TCPNAME queue manager attribute and the channel's LOCLADDR attribute supports IPv6. If the channel's CONNAME attribute resolves to an IPv4 address, then ensure the stack being used by the combination of the TCPNAME queue manager attribute and the channel's LOCLADDR attribute supports IPv4.

CSQX262E

csect-name Communications canceled, channel *channel-name* TRPTYPE=*trptype*

Severity
8

Explanation

An unexpected communications error occurred for a listener or a channel. This error occurs if the channel was stopped with mode FORCE and the communications session was canceled.

The channel is *channel-name*; in some cases its name cannot be determined and so is shown as '????'. *trptype* shows the communications system used:

TCP
TCP/IP

LU62
APPC/MVS

System action

The channel is stopped.

System programmer response

Restart the channel if appropriate.

CSQX272E

csect-name Thread mutex semaphore error

Severity
8

Explanation

The process attempted an operation on a thread mutex semaphore. The most likely cause of this problem is a shortage of an operating system resource (for example: memory).

System action

Processing continues.

System programmer response

Use any previous FFSTs to determine the reason for the failure. If you are unable to solve the problem, contact IBM Support.

CSQX293I

csect-name Channel *channel-name* has initiated a switch of transmission queue from *old-xmitq* to *new-xmitq*

Severity
0

Explanation

A switch of transmission queue for the channel identified by *channel-name* is required due to a change to the default cluster transmission queue configuration of the queue manager, or to the cluster channel name attribute of a cluster transmission queue. This message is issued by the channel initiator when the process of switching the transmission queue from *old-xmitq* to *new-xmitq* is started.

System action

The queue manager is notified to start the switching process for the channel.

The channel continues to run after closing the old transmission queue and switching to use the new transmission queue instead.

System programmer response

None.

CSQX294E

csect-name Transmission queue status unavailable, channel *channel-name*

Severity
8

Explanation

The transmission queue for the cluster-sender channel identified by *channel-name* cannot be determined because when the queue manager started it was unable to load the persisted transmission queue state from the queue SYSTEM.CHANNEL.SYNCQ.

System action

The channel ends abnormally.

System programmer response

If the queue manager is unable to load the persisted transmission queue state during startup it issues message CSQM561E.

CSQX296E

csect-name Password protection negotiation failed for channel *channel-name*, connection *conn-id*

Severity
8

Explanation

The channel *channel-name* could not be established because it failed to agree a password protection algorithm with the remote machine *conn-id*.

System action

The channel does not start.

System programmer response

Check whether password protection settings prevent interoperability with the remote machine.

Alternatively, consider using SSL or TLS to protect passwords instead. You must use a non-null CipherSpec to protect passwords.

CSQX298E

csect-name AMS not available, required for channel *channel-name*

Severity
8

Explanation

SPLPROT values of REMOVE and ASPOLICY require AMS to be active for this queue manager. If these values are specified without AMS being active, the channel cannot be started.

System action

The channel does not start.

System programmer response

Ensure AMS is active for this queue manager or alter the channel to SPLPROT(PASSTHRU), and try again.

CSQX403I

csect-name Auto-definition of channel *channel-name* suppressed by exit *exit-name*

Severity
0

Explanation

In response to a request to start a channel that was not defined, an attempt was made to define it automatically. The channel auto-definition exit *exit-name* prevented it being defined.

System action

The channel is not started.

CSQX404I

csect-name Phase one of REFRESH CLUSTER REPOS(YES) has completed, cluster *cluster_namen* objects changed

Severity
0

Explanation

Phase one of REFRESH CLUSTER has completed.

Applications attempting to access cluster resources may see failures to resolve cluster resources until phase two of REFRESH CLUSTER is complete.

Phase two is complete once all new information has been received from other members of the cluster.

Monitor your SYSTEM.CLUSTER.COMMAND.QUEUE to determine when it has reached a consistently empty state to indicate that the refresh process has completed.

System action

None.

CSQX405I

csect-name FORCEREMOVE QUEUES(YES) command processed, cluster *cluster_name* target *target*

Severity
0

Explanation

The repository manager successfully processed a RESET CLUSTER ACTION(FORCEREMOVE) command with the QUEUES(YES) option for the indicated cluster and target queue manager.

System action

None.

CSQX406E

csect-name REFRESH CLUSTER REPOS(YES) command failed, cluster *cluster_name* - *qmgr-name* is a full repository

Severity
8

Explanation

The repository manager could not process a REFRESH CLUSTER command with the REPOS(YES) option for the indicated cluster, because the local queue manager provides full repository management service for the cluster.

System action

The command is ignored.

System programmer response

Reissue the command with the correct values or on the correct queue manager. It might be necessary to change the queue manager so that it is not a full repository for the cluster.

CSQX407I

csect-name Cluster queue *q-name* definitions inconsistent

Severity
4

Explanation

The definition of a cluster queue has different values for the DEFPRTY, DEFPSIST, DEFPRESP, and DEFBIND attributes on the various queue managers in the cluster.

All definitions of the same cluster queue must be identical. Problems might arise if your applications rely on one of these attributes to determine messaging behavior. For example, if an application opens a cluster queue with the option MQOO_BIND_AS_Q_DEF, and the different instances of the queue have different DEFBIND values, the behavior of the message transfer depends on which instance of the queue happens to be selected when it is opened.

System action

None.

System programmer response

Alter the definitions of the queue on the various queue managers so that they have identical values for these attributes. Any inconsistent definitions in the cluster (flagged by message CSQX407I) can be displayed by issuing the following command (from a full repository) `DISPLAY QCLUSTER(q-name) ALL`.

CSQX410I

csect-name Repository manager started

Severity
0

Explanation

The repository manager started successfully.

System action

None.

CSQX411I

csect-name Repository manager stopped

Severity
0

Explanation

The repository manager stopped. This may be for one of three reasons:

- The channel initiator is stopping.
- The channel initiator is starting and the queues used by the repository manager have not been defined because clustering is not required.
- An error has occurred.

System action

Processing continues, but clustering is not available.

System programmer response

If an error has occurred, investigate the problem reported in the preceding messages.

CSQX412E

csect-name Misdirected repository command, target *target-id* sender *sender-id*

Severity
8

Explanation

The repository manager received a command intended for some other queue manager, with an identifier that is *target-id*. The command was sent by the queue manager with identifier *sender-id*.

System action

The command is ignored, and the error is reported to the sender.

System programmer response

Check the channel and cluster definitions of the sending queue manager.

CSQX413E

csect-name Repository command format error, command code *command*

Severity
8

Explanation

An internal error has occurred.

System action

The command is ignored, and the error is reported to the sender; the repository manager continues processing. Information about the error is written to the data set identified by the CSQSNAP DD statement of the channel initiator started task JCL procedure, xxxxCHIN.

System programmer response

Collect the items listed in the Problem Determination section and contact IBM Support.

CSQX415E

csect-name Repository command state error, command code *command* cluster object *object-name* sender *sender-id*

Severity
8

Explanation

An internal error has occurred.

System action

The command is ignored; the repository manager continues processing. Information about the error is written to the data set identified by the CSQSNAP DD statement of the channel initiator started task JCL procedure, xxxxCHIN.

System programmer response

Collect the items listed in the Problem Determination section and contact IBM Support.

CSQX416E

csect-name Repository command processing error, RC=*return-code* command code *command* cluster object *object-name* sender *sender-id*

Severity
8

Explanation

An internal error has occurred.

System action

The command is ignored; the repository manager continues processing. Information about the error is written to the data set identified by the CSQSNAP DD statement of the channel initiator started task JCL procedure, xxxxCHIN.

System programmer response

Collect the items listed in the Problem Determination section and contact IBM Support.

CSQX417I

csect-name Cluster-senders remain for removed queue manager *qmgr-name*

Severity
0

Explanation

The indicated queue manager has been deleted or forcibly removed from a cluster, but there are manually-defined cluster-sender channels that refer to it. This means that the repository manager will continue to send cluster information to the removed queue manager.

System programmer response

Delete the manually-defined cluster-sender channels that refer to *qmgr-name*.

CSQX418I

csect-name Only one repository for cluster *cluster_name*

Severity
0

Explanation

The repository manager has received information about a cluster for which it is the only full repository.

System action

None.

System programmer response

If you require a second full repository, alter the REPOS or REPOSNL attribute of the second queue manager that is to have a full repository for the cluster to specify the cluster name.

CSQX419I

csect-name No cluster-receivers for cluster *cluster_name*

Severity
0

Explanation

The repository manager has received information about a cluster for which no cluster-receiver channels are known.

System action

None.

System programmer response

Define cluster-receiver channels for the cluster on the local queue manager.

CSQX420I

csect-name No repositories for cluster *cluster_name*

Severity
0

Explanation

The repository manager has received information about a cluster for which no full repositories are known.

System action

None.

System programmer response

Define a cluster-sender channel for connecting to the queue manager that is the full repository for the cluster, or alter the REPOS or REPOSNL attribute of the queue manager that is to have a full repository for the cluster to specify the cluster name.

CSQX422E

csect-name Repository manager error, RC=*return-code*

Severity
8

Explanation

An internal error has occurred.

System action

The repository manager attempts to continue processing. Information about the error is written to the data set identified by the CSQSNAP DD statement of the channel initiator started task JCL procedure, xxxxCHIN.

System programmer response

Collect the items listed in the Problem Determination section and contact IBM Support.

CSQX425E

csect-name Repository command merge error, command code *command* cluster object *object-name* sender *sender-id*

Severity
8

Explanation

An internal error has occurred.

System action

The command is ignored; the repository manager continues processing. Information about the error is written to the data set identified by the CSQSNAP DD statement of the channel initiator started task JCL procedure, xxxxCHIN.

System programmer response

Collect the items listed in the Problem Determination section and contact IBM Support.

CSQX426E

csect-name Undeliverable repository command, channel *channel-name* target *target-id* command code *command*

Severity
8

Explanation

The repository manager tried to send a command to another queue manager using channel *channel-name*. The other queue manager, with identifier *target-id*, could not be found.

System action

The command is ignored.

System programmer response

Check the channel and cluster definitions of the sending and receiving queue managers.

CSQX427E

csect-name Cluster-sender not connected to repository, cluster *cluster_name* channel *channel-name* target *target-id*

Severity
8

Explanation

A cluster-sender channel must be connected to a queue manager that is a full repository for all the clusters for the channel, and the corresponding cluster-receiver channel must be in the same clusters. Channel *channel-name* in cluster *cluster_name* does not satisfy this. *target-id* is the identifier of the target queue manager for the channel.

System action

The command is ignored.

System programmer response

Check the definition of the channel on both queue managers to ensure that it is connected to a full repository for the clusters, and that it is in the same clusters on both queue managers.

CSQX428E

csect-name Unexpected publication of a cluster queue, cluster *cluster_name* cluster queue *q-name* sender *sender-id*

Severity
8

Explanation

The repository manager received a publication for cluster queue *q-name* from another queue manager, with an identifier *sender-id*, relating to cluster *cluster_name*. The local queue manager cannot accept the command because it is not a full repository for the cluster and thus it does not have an interest in the cluster queue.

This can also occur because a command destined for the local repository manager is delayed in the network and is out of date when it arrives, for example because a REFRESH CLUSTER command has been issued on the local repository manager and caused its view of the cluster to change.

System action

The command is ignored.

System programmer response

If the local partial repository queue manager is supposed to be a full repository for the cluster, use the ALTER QMGR command to specify a repository or repository namelist which contains the cluster. If the local queue manager is correctly a partial repository for the cluster, ensure that the remote queue manager does not have a manually defined cluster sender directed at the local partial repository.

If the message occurs because a command is out of date, the message can be ignored.

CSQX429E

csect-name Unexpected deletion of a cluster queue, cluster *cluster_name* cluster queue *q-name*

Severity
8

Explanation

The repository manager received a deletion for cluster queue *q-name* from another queue manager, with an identifier *sender-id*, relating to cluster *cluster_name*. The local queue manager cannot accept the command because it is not a full repository for the cluster and thus it does not have an interest in the cluster queue.

This can also occur because a command destined for the local repository manager is delayed in the network and is out of date when it arrives, for example because a REFRESH CLUSTER command has been issued on the local repository manager and caused its view of the cluster to change.

System action

The command is ignored.

System programmer response

If the local partial repository queue manager is supposed to be a full repository for the cluster, use the ALTER QMGR command to specify a repository or repository namelist which contains the cluster. If the local queue manager is correctly a partial repository for the cluster, ensure that the remote queue manager does not have a manually defined cluster sender directed at the local partial repository.

If the message occurs because a command is out of date, the message can be ignored.

CSQX430E

csect-name Unexpected queue manager repository command, cluster *cluster_name* channel *channel-name* sender *sender-id*

Severity
8

Explanation

The repository manager received a command from another queue manager, with an identifier that is *sender-id*, relating to cluster *cluster_name*. The local queue manager cannot accept the command because it is not a full repository for the cluster, it does not have an interest in the cluster channel, and it does not have any matching cluster-sender channels. The cluster-sender channel used by the other queue manager was *channel-name*.

This message might appear on a queue manager that has defined a cluster-sender channel to another queue manager that does not host a full repository, if the other queue manager is later modified to host a full repository.

System action

The command is ignored.

System programmer response

Check the definition of the channel on the sending queue manager to ensure that it is connected to a full repository for the cluster.

Ensure the CLUSTER and CLUSNL values are consistent, and that you have not specified a *cluster_name* when you meant a *cluster-namelist*.

CSQX431I

csect-name Repository unavailable, cluster *cluster_name* channel *channel-name* sender *sender-id*

Severity
0

Explanation

The repository manager received a command from another queue manager, with identifier *sender-id*, reporting that it is no longer a full repository for cluster *cluster_name*.

System action

The cluster-sender channel *channel-name* is changed so that it can no longer be used to access the other queue manager in relation to the cluster.

CSQX432I

csect-name Unexpected cluster query received, cluster *cluster_name* cluster object *object-name* sender *sender-id*

Severity
8

Explanation

The repository manager received a query for cluster object *object-name* from another queue manager, with an identifier *sender-id*, relating to cluster *cluster_name*. The local queue manager cannot accept the command because it is not a full repository for the cluster.

This can also occur because a command destined for the local repository manager is delayed in the network and is out of date when it arrives, for example because a REFRESH CLUSTER command has been issued on the local repository manager and caused its view of the cluster to change.

System action

The command is ignored.

System programmer response

If the local partial repository queue manager is supposed to be a full repository for the cluster, use the ALTER QMGR command to specify a repository or repository namelist which contains the cluster. If the local queue manager is correctly a partial repository for the cluster, ensure that the remote queue manager does not have a manually defined cluster sender directed at the local partial repository.

If the message occurs because a command is out of date, the message can be ignored.

CSQX433E

csect-name Cluster-receiver and cluster-sender differ, cluster *cluster_name* channel *channel-name* sender *sender-id*

Severity
8

Explanation

The repository manager received a command from another queue manager, with identifier *sender-id*. The cluster-sender channel *channel-name* on that queue manager is in cluster *cluster_name*, but the corresponding cluster-receiver channel on the local queue manager is not.

System action

The command is ignored.

System programmer response

Change the definition of the channel so that it is in the same clusters on both queue managers.

CSQX434E

csect-name Unrecognized message on *name*

Severity
8

Explanation

The channel initiator found a message on one of its queues that either had a format that could not be recognized or did not come from a queue manager or channel initiator.

System action

The message is put on the dead-letter queue.

System programmer response

Examine the message on the dead-letter queue to determine the originator of the message.

CSQX435E

csect-name Unable to put repository manager message, target *target-id* MQCC=*mqcc* MQRC=*mqrc* (*mqrc-text*)

Severity
4

Explanation

The repository manager tried to send a message to SYSTEM.CLUSTER.COMMAND.QUEUE on another queue manager with an identifier that is *target-id*, but the MQPUT call was unsuccessful.

System action

Processing continues, but repository information may be out of date.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form).

Check the channel and cluster definitions on the local and target queue managers, and ensure that the channels between them are running.

When the problem is corrected, the repository information will normally be updated automatically. The REFRESH CLUSTER command can be used to be sure that the repository information is up to date.

This error may occur if the REFRESH CLUSTER REPOS(YES) command is issued against a full repository, as the full repository will then be temporarily unable to fulfil requests from other repositories until it has rebuilt the cluster. If there is more than one full repository for the cluster, the problem will resolve itself. If there is only a single full repository for the cluster, the REFRESH CLUSTER command will need to be run against all the other queue managers in the cluster to make them contact the full repository again.

CSQX436E

csect-name Unable to put repository manager message, cluster *cluster_name* MQCC=*mqcc* MQRC=*mqrc* (*mqrc-text*)

Severity
4

Explanation

The repository manager tried to send a message to SYSTEM.CLUSTER.COMMAND.QUEUE on a queue manager that has the full repository for the specified cluster, but the MQPUT was unsuccessful.

System action

Processing continues, but repository information may be out of date.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form).

Check the channel and cluster definitions on the local and target queue managers, and ensure that the channels between them are running.

When the problem is corrected, the repository information will normally be updated automatically. The REFRESH CLUSTER command can be used to be sure that the repository information is up to date.

CSQX437E

csect-name Unable to commit repository changes

Severity
4

Explanation

The repository manager tried to commit some updates to the repository but was unsuccessful.

System action

Processing continues, but local repository information might be out of date.

System programmer response

If this occurs when the channel initiator is stopping, it can be ignored because the local repository information will normally be updated automatically when the channel initiator is restarted. If there is an isolated occurrence at other times, use the REFRESH CLUSTER command to bring the local repository information up to date.

If the problem persists, contact IBM Support.

CSQX438E

csect-name Unable to reallocate messages, channel *channel-name* MQCC=*mqcc* MQRC=*mqrc* (*mqrc-text*)

Severity
8

Explanation

The repository manager was unable to reallocate messages for the specified channel to another destination.

System action

The messages remain on the transmission queue.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form).

Use this information in conjunction with any preceding error messages to determine the cause of the problem. When the problem is corrected, restart the channel.

CSQX439E

csect-name Repository error for channel *channel-name*

Severity
8

Explanation

An internal error has occurred.

System action

The repository manager attempts to continue processing. Information about the error is written to the data set identified by the CSQSNAP DD statement of the channel initiator started task JCL procedure, xxxxCHIN.

System programmer response

Collect the items listed in the Problem Determination section and contact IBM Support.

CSQX440E

csect-name FORCEREMOVE command failed, cluster *cluster_name* target *target* - repository is not on *qmgr-name*

Severity
8

Explanation

The repository manager could not process a RESET CLUSTER ACTION(FORCEREMOVE) command for the indicated cluster and target queue manager, because the local queue manager does not provide a full repository management service for the cluster.

System action

The command is ignored.

System programmer response

Reissue the command with the correct values or on the correct queue manager.

CSQX441I

csect-name FORCEREMOVE command processed, cluster *cluster_name* target *target*

Severity
0

Explanation

The repository manager successfully processed a RESET CLUSTER ACTION(FORCEREMOVE) command for the indicated cluster and target queue manager.

System action

None.

CSQX442I

csect-name Phase one of REFRESH CLUSTER has completed, cluster *cluster_namen* objects changed

Severity
0

Explanation

Phase one of REFRESH CLUSTER has completed.

Applications attempting to access cluster resources may see failures to resolve cluster resources until phase two of **REFRESH CLUSTER** is complete.

Phase two is complete once all new information has been received from other members of the cluster.

Monitor your SYSTEM.CLUSTER.COMMAND.QUEUE to determine when it has reached a consistently empty state to indicate that the refresh process has completed.

System action

None.

CSQX443I

csect-name SUSPEND QMGR command processed, cluster *cluster_namen* objects changed

Severity
0

Explanation

The repository manager successfully processed a SUSPEND QMGR command for the indicated cluster. (Where the command specified a namelist of clusters, the message is issued only for the first cluster in the namelist.)

System action

None.

CSQX444I

csect-name RESUME QMGR command processed, cluster *cluster_namen* objects changed

Severity
0

Explanation

The repository manager successfully processed a RESUME QMGR command for the indicated cluster. (Where the command specified a namelist of clusters, the message is issued only for the first cluster in the namelist.)

System action

None.

CSQX447E

csect-name Unable to backout repository changes

Severity
8

Explanation

Following an error, the repository manager tried to backout some updates to the local repository but was unsuccessful.

System action

The repository manager terminates.

System programmer response

If the repository manager subsequently restarts successfully, or if on restarting the channel initiator the repository manager subsequently starts successfully, this can be ignored.

If not, contact IBM Support.

CSQX448E

csect-name Repository manager stopping because of errors. Restart in *n* seconds

Severity
8

Explanation

A severe error, as reported in the preceding messages, occurred during repository manager processing; the repository manager is unable to continue.

System action

The repository manager terminates. The channel initiator will try to restart it after the specified interval.

System programmer response

Correct the problem reported in the preceding messages.

CSQX449I

csect-name Repository manager restarted

Severity
0

Explanation

The repository manager restarted successfully following an error.

System action

None.

CSQX453E

csect-name FORCEREMOVE command failed, cluster *cluster_name* target *target* is not unique

Severity
8

Explanation

The repository manager could not process a RESET CLUSTER ACTION(FORCEREMOVE) command for the indicated cluster and target queue manager, because there is more than one queue manager with the specified name in the cluster.

System action

The command is ignored.

System programmer response

Reissue the command specifying the identifier (QMID) of the queue manager to be removed, rather than its name.

CSQX454E

csect-name REFRESH CLUSTER REPOS(YES) command failed, because channel & could not be stopped.

Severity

8

Explanation

The queue manager could not fully process the REFRESH CLUSTER REPOS(YES) command because the named channel could not be stopped within a reasonable time.

The rest of the REFRESH CLUSTER REPOS(YES) processing has been abandoned. At the time this error message is written, the named channel is in an unknown state, but this state might have changed by the time you see this message. If the queue manager changed any channels to STOPPED state, then the queue manager now attempts to change them into INACTIVE state so that they may start automatically.

Any channels that were already in STOPPED state when the REFRESH CLUSTER REPOS(YES) command began are left unchanged. A copy of this error message is written once for each channel that failed to stop within a reasonable time.

System action

Refresh cluster processing is abandoned.

System programmer response

Check the state of the named channel and all other cluster channels, in case they have been left in STOPPED state, and this is not your desired state for them.

If there are any problems that are evident for the named channel (for example, network problems outside of IBM MQ), then address these problems before attempting the REFRESH CLUSTER REPOS(YES) command again.

To be sure the command will not hit similar problems the next time you run it, then you should manually change all relevant cluster channels into STOPPED state before attempting the REFRESH CLUSTER REPOS(YES) command again.

CSQX455E

csect-name FORCEREMOVE command failed, cluster *cluster_name* target *target* not found

Severity

8

Explanation

The repository manager could not process a RESET CLUSTER ACTION(FORCEREMOVE) command for the indicated cluster and target queue manager, because no information about that queue manager was found in the local repository.

System action

The command is ignored.

System programmer response

Reissue the command specifying the correct queue manager name or identifier.

CSQX456E

csect-name Full repository update not received, cluster *cluster_name* cluster object *object_name* (queue manager *qmgr_uuid*)

Severity

4

Explanation

The repository manager found a cluster object that had been used in the last 30 days, and for which updated information should have been received. However, no such information has been received. The cluster object is *object_name* in *cluster_name*, and its queue manager is *qmgr_uuid*.

If the queue manager is a partial repository for the object, the updated information should have been sent from a full repository. If the queue manager is a full repository, the updated information should have been sent from the queue manager on which the object is defined.

System action

The repository manager keeps information about this object for a further 60 days from when the error first occurred. If information has not been sent to a full repository then this object is not used to satisfy any new requests for cluster resources made to this full repository.

System programmer response

If the cluster object is still required, check that:

- The cluster channels to and from the queue manager that is the full repository for the cluster, and between there and the queue manager where the object is located, are able to run.
- The repository managers on those queue managers have not ended abnormally.
- There is not a long-running problem with the cluster receiver channel to the local queue manager in cluster *cluster_name*. If there is a problem, correct the problem urgently, to ensure that updates for the cluster are received.
- There is not a long-running problem on the cluster sender channel of the remote queue manager in cluster *cluster_name*. If there is a problem, correct the problem urgently, to ensure that updates for the cluster are sent.
- The remote queue manager is not out of step with this queue manager, potentially due to a restore of the remote queue manager from a backup. If the remote queue manager is out of step, issue a REFRESH CLUSTER command on the remote queue manager to synchronize with other queue managers in the cluster.
- The remote queue manager is not out of step with this queue manager, potentially due to a disaster recovery exercise in which a replacement queue manager with the same cluster receiver channel name was created, was run for a while, then ended. If this has happened, then the remote queue manager *qmgr_uuid* must now issue a REFRESH CLUSTER command to synchronize with other queue managers in the cluster.
- If the above items have been checked, and this problem persists over several days (causing repeats of this error message in the error logs of the local queue manager) contact your IBM support center.

CSQX457I

csect-name Repository available, cluster *cluster_name* channel *channel-name* sender *sender-id*

Severity

0

Explanation

The repository manager received a command from another queue manager, with identifier *sender-id*, reporting that it is once again a full repository for cluster *cluster_name*.

System action

The cluster-sender channel *channel-name* is changed so that it can be used to access the other queue manager in relation to the cluster.

CSQX458E

csect-name Unable to access repository cache exclusively, TCB= *tcb-name* has *num-registrations* outstanding registrations

Severity

8

Explanation

During an operation that requires exclusive access to the cache, another task was found to be registered. If the queue manager finds registrations still exist after waiting for the task to remove its registrations, the queue manager issues this message. The task preventing exclusive access to the repository cache has *num-registrations* outstanding registrations.

System action

Processing continues.

System programmer response

Determine if this task is still running or terminated. If the task is not running or if the problem persists collect the items listed in the Problem determination on z/OS section and contact IBM Support.

CSQX459E

csect-name Cluster topic topic-name from qmgr-name rejected due to PSCLUS(DISABLED)

Severity

8

Explanation

Information regarding cluster topic topic-name has been sent to this queue manager over a channel from qmgr-name but the queue manager attribute PSCLUS has been set to DISABLED, indicating that Publish/Subscribe activity is not expected between queue managers in this cluster.

System action

The cluster topic definition is ignored and will not be visible from this queue manager.

System programmer response

To enable publish/subscribe clustering, alter the PSCLUS attribute on all queue managers in the cluster to ENABLED. You may also need to issue REFRESH CLUSTER and REFRESH QMGR commands as detailed in the documentation for the PSCLUS attribute. If you are not using publish/subscribe clusters you should delete the clustered topic object, and ensure PSCLUS is DISABLED on all queue managers.

CSQX460E

csect-name Cluster cache is full

Severity

8

Explanation

No more space is available in the cluster cache area.

System action

The repository manager terminates. The channel initiator will try to restart it after the specified interval.

System programmer response

The problem may be temporary. If it persists, the queue manager must be restarted; this will cause more space to be allocated for the cluster cache area.

Consider changing the cluster cache type system parameter CLCACHE to dynamic, so that more space for the cache will be obtained automatically as required. (If you are using a cluster workload exit, ensure that it supports a dynamic cluster cache.) For information about the system parameters for the CSQ6SYSP macro, see Using CSQ6SYSP.

CSQX461I

csect-name Cluster cache entry corrected, cluster queue manager *clusqmgr-name* channel *channel-name* connection *conn-id*

Severity
4

Explanation

At channel initiator restart, the repository manager found a corrupted entry in the cluster cache. The entry has been corrected.

System action

Processing continues. The cluster channel to which the entry refers, *channel-name* using connection *conn-id*, will be available for use.

System programmer response

None. You can verify that the entry was successfully corrected by issuing the command DISPLAY CLUSQMGR(*clusqmgr-name*) on the queue manager where this message was issued.

CSQX462E

csect-name Cluster cache entry is unusable, cluster queue manager *clusqmgr-name* channel *channel-name* connection *conn-id*

Severity
8

Explanation

At channel initiator restart, the repository manager found a corrupted entry in the cluster cache which could not be corrected.

System action

The corrupted entry is ignored. The cluster channel to which it refers, *channel-name* using connection *conn-id*, will not be usable.

System programmer response

The corrupted entry must be corrected and reintroduced by issuing the command

```
ALTER CHANNEL(channel-name) CHLTYPE(CLUSRCVR)
```

on the cluster queue manager *clusqmgr-name*. You can verify that the entry was successfully reintroduced by issuing the command DISPLAY CLUSQMGR(*clusqmgr-name*) on the queue manager where this message was issued.

CSQX463E

csect-name Error accessing cluster cache entry

Severity
8

Explanation

There was an internal error when accessing a cluster cache entry.

System action

Information about the error is written to the data set identified by the CSQSNAP DD statement of the channel initiator started task JCL procedure, xxxxCHIN. The component where the error occurred (message channel agent, repository manager) usually terminates; in some cases, the end result will be that the channel initiator terminates.

System programmer response

Collect the items listed in the Problem Determination section and contact IBM Support.

CSQX465I

csect-name New cluster topic definition inconsistent, topic *topic-name*, queue manager identifier *qmid*, attribute *attr*

Severity
4

Explanation

The definition of the cluster topic *topic-name*, defined on queue manager identifier *qmid* has different *attr* attribute values than one or more cluster topics that already exist in the cluster cache. The existing topic objects are reported by message CSQX466I.

All definitions of the same cluster topic should be identical; otherwise, problems may arise if your applications rely on one of these attributes to determine messaging behavior. For example, if an application opens a cluster topic and the different instances of the topic have different TOPICSTR values, the behavior of the message transfer depends on which instance of the topic happens to be selected when it is opened.

System action

None.

System programmer response

Alter the definitions of the topic on the various queue managers so that they have identical values for all attributes.

CSQX466I

csect-name Cluster topic definitions inconsistent, topic *topic-name*, queue manager identifier *qmid* attribute *attr*

Severity
4

Explanation

The definition of the cluster topic *topic-name*, defined on queue manager identifier *qmid* has different *attr* attribute value than a cluster topic being added to the cluster cache. The topic object being added is reported by message CSQX465I.

All definitions of the same cluster topic should be identical; otherwise, problems may arise if your applications rely on one of these attributes to determine messaging behavior. For example, if an application opens a cluster topic and the different instances of the topic have different TOPICSTR values, the behavior of the message transfer depends on which instance of the topic happens to be selected when it is opened.

System action

None.

System programmer response

Alter the definitions of the topic on the various queue managers so that they have identical values for all attributes.

CSQX467E

Repository error for topic *topic-name*, MQCC=*mqcc* MQRC=*mqrc* (*mqrc-text*)

Severity
8

Explanation

The cluster repository was unable to insert or delete topic *topic-name* due to an unexpected error in the queue manager.

System action

The repository manager terminates. The channel initiator tries to restart the repository manager after an interval. See message CSQX448E for more information.

System programmer response

For more information about *mqcc* and *mqrc* completion codes (*mqrc-text* provides the MQRC in textual form), see "API completion and reason codes" on page 1151.

Contact IBM Support with the reason code provided for this failure.

CSQX468I

csect-name Queue manager *qmgr-uuid1* has replaced queue manager *qmgr-uuid2* in a cluster due to reuse of channel *channel-name*

Severity
0

Explanation

Queue manager *qmgr-uuid1* has joined a cluster using a cluster receiver channel with the same name as one that has already been defined by queue manager *qmgr-uuid2*. All cluster receiver channels used within a cluster must be uniquely named.

System action

Queue manager *qmgr-uuid1* uses channel *channel-name*. Queue manager *qmgr-uuid2* cannot successfully participate in the cluster while queue manager *qmgr-uuid1* is a member.

System programmer response

The use of a channel name currently associated with a different queue manager in the cluster can be intentional, for example it is possible the original queue manager has been deleted and re-created as a new queue manager. However, accidental duplication of a channel name across multiple queue managers would also result in this behavior. If this action was not intended review the configuration of the queue managers.

CSQX469E

csect-name Update not received for CLUSRCVR channel *channel-name* hosted on queue manager *qmid* in cluster *cluster_name*, expected *n* days ago, *m* days remaining

Severity
8

Explanation

The repository manager detected that the CLUSRCVR channel has not been republished by its owning queue manager. This republish action should have happened automatically *n* days ago, or in the time between then and now.

System action

The repository manager will check for this condition approximately every hour, continuing for a period of approximately *m* days from now. If an update for the CLUSRCVR channel is received during this period, these messages will stop. If no update is received, these messages will continue to be written. However, after this period has elapsed, if no update has been received, the local queue manager will discard its knowledge of this channel, and these messages will stop. You should be aware that Partial Repository queue managers in this cluster will cease to be able to use the channel at about that time.

System programmer response

There are several possible responses:

1. If the channel had been removed intentionally, and is no longer required, you should consider removing it fully via the RESET CLUSTER command.
2. There is a long-running problem with the local queue manager's CLUSRCVR in cluster *cluster_name*. If this is true, then correct the problem urgently, to ensure that updates for the cluster are received.
3. There is a long-running problem on the remote queue manager's CLUSSDR in cluster *cluster_name*. If this is true, then correct the problem urgently, to ensure that updates for the cluster are sent.
4. Check that the repository manager on the remote queue manager has not ended abnormally.
5. The remote queue manager is out of step with this queue manager, potentially due to a restore of the queue manager from a backup, potentially due to a restore of either the local or remote queue manager from backup data, including situations in which a duplicate instance of a queue manager has been introduced to the cluster even if only temporarily, for example under disaster recovery testing.

   See Clustering: Availability, multi-instance, and disaster recovery for more information, and the affected queue manager must issue REFRESH CLUSTER to synchronize with other queue managers in the cluster.

If the above items have been checked, and this problem persists over several days, causing repeats of this error message in the local queue manager's error logs, contact IBM Support.

CSQX470E

csect-name Channel *channel-name* has the wrong disposition *disposition*

Severity

8

Explanation

The action you requested cannot be performed on channel *channel-name* because it has the wrong disposition. For example, the action asked for a shared channel, but its disposition is private.

System action

The requested action is not performed.

System programmer response

Check whether the channel name is specified correctly. If it is, check that:

- The channel has been defined correctly
- The transmission queue name identifies the correct queue, and that queue has the required disposition.

The disposition of an instance of a channel is **not** related to that specified by QSGDISP in the channel definition:

- A sending channel is *shared* if its transmission queue is shared, and *private* if it is not.
- A receiving channel is *shared* if it was started in response to an inbound transmission directed to the queue sharing group, and *private* if it was started in response to an inbound transmission directed to the queue manager.

CSQX471I

csect-name nn shared channels to restart, *nn* requests issued

Severity
0

Explanation

The channel initiator is shutting down; it owns some active shared sending channels, and they have not been requested to stop. Requests to restart these channels on another queue manager have been issued as shown.

System action

The channel initiator shutdown processing continues.

System programmer response

If the numbers in the message differ, the channel initiator was not able to issue restart requests for all the channels. In this case, use the DISPLAY CHSTATUS command to determine which channels are still owned by the queue manager for the channel initiator that is shutting down, and which therefore have not been restarted, and restart them manually as required.

CSQX475I

csect-name Channel *channel-name* adopted, connection *conn-id*

Severity
0

Explanation

Channel *channel-name*, which was orphaned because of a communications error, has been adopted by a new instance of the channel, from connection *conn-id*.

System action

Processing continues.

CSQX476E

csect-name Channel *channel-name* is active on *qmgr-name*, shared status entry found

Severity
8

Explanation

An operation was requested on a channel that is active. Because the channel is shared, it might be active on another queue manager. If the channel is a receiver, a previous instance of it might have been orphaned and therefore still be active.

System action

The request fails.

System programmer response

For operations other than starting the channel, either stop the channel manually, or wait for it to terminate, and try the operation again. It might be necessary to use MODE(FORCE) to stop the channel manually if the Adopt MCA function is not being used. Using the Adopt MCA function avoids the need for manual intervention to handle orphaned receiver channels.

If the channel is not running on the named queue manager, then there is an orphaned shared status entry, which might be because a loss of connectivity to Db2 occurred. If the problem persists, contact IBM Support.

CSQX477E

csect-name Channel *channel-name* is active, transmission queue *queue-name* in use on *qmgr-name*

Severity
8

Explanation

An operation was requested on a channel that is active. While starting, a sender channel has detected that its transmission queue, *queue-name*, is already in use on the specified queue manager.

System action

The request fails.

System programmer response

Do the following, which might need to be done on a different queue manager other than the one the channel was started on, as appropriate:

- Check if the channel is already running
- Check if another channel is using the queue by using the DISPLAY QSTATUS command
- Ensure the queue name is specified correctly in the channel definition
- Alter the queue usage attribute of the queue to that of a transmission queue.

If the channel is already running, for operations other than starting the channel, either stop the channel manually, or wait for it to terminate, and retry the operation. It may be necessary to use MODE(FORCE) to stop the channel manually if the Adopt MCA function is not being used. Using the Adopt MCA function will avoid the need for manual intervention to handle orphaned receiver channels.

CSQX478E

csect-name Channel *channel-name* is active on *qmgr-name*, connection tag in use

Severity
8

Explanation

An operation was requested on a channel that is active. The connection tag used to serialize the channel within the queue sharing group is currently in use. Because the channel is shared, it might be active on another queue manager. If the channel is a receiver, a previous instance of it might have been orphaned and therefore still be active.

In addition to the CSQX478E for a shared channel, another possible symptom is CSQX514E: *csect-name* Channel channel-name is active on qmgr-name. The new instance of the channel is starting with a different IP address from the running instance. If the sender's IP address changed or might translate into more than one address, set ADOPTCHK to QMNAME using the ALTER QMGR command. For example, /cpf ALTER QMGR ADOPTCHK(QMNAME) where "cpf" is the command prefix for the queue manager subsystem.

System action

The request fails.

System programmer response

For operations other than starting the channel, either stop the channel manually, or wait for it to terminate, and try the operation again. It might be necessary to use MODE(FORCE) to stop the channel manually if the Adopt MCA function is not being used. Using the Adopt MCA function avoids the need for manual intervention to handle orphaned receiver channels.

CSQX479E

csect-name Channel *channel-name* is active on *qmgr-name*, shared channel adoption failed

Severity
8

Explanation

An attempt was made to adopt channel *channel-name*, which was orphaned because of a communications error. It failed, either because the channel could not be stopped or because a response was not received from the queue manager *qmgr-name*.

System action

The request fails, and the orphaned channel might remain active.

System programmer response

Investigate any preceding error messages to discover why the adopt failed. Either stop the channel manually, or wait for it to terminate, and try the operation again. It might be necessary to use MODE(FORCE) to stop the channel manually.

CSQX482E

csect-name Shared channel function not available

Severity
8

Explanation

During the execution of a channel command, or during shared channel processing, an internal function required by the channel initiator was found to be unavailable.

System action

The channel command fails or the channel stops.

System programmer response

Check that the Db2 tables required by IBM MQ are correctly defined, and restart the queue manager and Db2 if necessary. If these appear to be running correctly, display the information in the shared channel status (CSQ.ADMIN_B_SCST) and the shared synchronization key (CSQ.ADMIN_B_SSKT) Db2 tables, and contact IBM Support for further assistance. For further information, and for details of a sample job (CSQ45STB) which shows the information in the Db2 tables, see Problem determination on z/OS.

CSQX483E

csect-name Db2 not available

Severity
8

Explanation

Because Db2 is not available, or is no longer available, the channel initiator cannot do processing for a shared channel.

System action

The channel command fails or the channel stops.

System programmer response

Use the preceding messages on the z/OS console to investigate why Db2 is not available, and restart it if necessary.

CSQX484E

csect-name Error accessing Db2

Severity
8

Explanation

Because there was an error in accessing Db2, the channel initiator cannot do processing for a shared channel.

System action

The channel command fails or the channel stops.

System programmer response

Resolve the error reported in the preceding messages.

CSQX485E

csect-name Shared channel status error

Severity
8

Explanation

During the execution of a channel command, or during shared channel processing, shared channel status or shared synchronization key information, held in Db2, was found to be corrupted.

System action

The channel command fails or the channel stops.

System programmer response

Check that the Db2 tables required by IBM MQ are correctly defined, and restart Db2 if necessary. If Db2 appears to be running correctly, display the information in the shared channel status (CSQ.ADMIN_B_SCST) and the shared synchronization key (CSQ.ADMIN_B_SSKT) Db2 tables, and contact IBM Support for further assistance. For further information, and for details of a sample job (CSQ45STB) which shows the information in the Db2 tables, see Problem determination on z/OS.

CSQX486E

csect-name Shared channel *channel-name* definitions inconsistent

Severity
8

Explanation

The definition of a shared channel has differing attribute values on the various queue managers in the queue sharing group. For example, if the type of the channel differs start or stop requests cannot operate correctly.

System action

The request fails.

System programmer response

Change the definitions of the channel so that they are the same on all the queue managers. If the channel type needs changing, you must delete and then redefine the channel.

CSQX489E

csect-name Maximum instance limit *limit* exceeded, channel *channel-name* connection *conn-id*

Severity
8

Explanation

There are too many instances of the channel *channel-name* running to be able to start another. The maximum number allowed is *limit* and is specified in the MAXINST channel attribute.

System action

The channel does not start.

System programmer response

Wait for some of the operating channels to terminate before restarting the channel, or use the ALTER CHANNEL command to increase MAXINST.

CSQX490E

csect-name Maximum client instance limit *limit* exceeded, channel *channel-name* connection *conn-id*

Severity
8

Explanation

There are too many instances of the channel *channel-name* running from the connection *conn-id* to be able to start another. The maximum number allowed is *limit* and is specified in the MAXINSTC channel attribute.

System action

The channel does not start.

System programmer response

Wait for some of the operating channels to terminate before restarting the channel, or use the ALTER CHANNEL command to increase MAXINSTC.

CSQX496I

csect-name Channel *channel-name* stopping because of request by remote exit

Severity
0

Explanation

The channel is closing because the user channel exit at the remote end requested it.

System action

The channel stops. The associated transmission queue might be set to GET(DISABLED) and triggering turned off. For auto-defined channels, the channel does not start.

System programmer response

Note that this puts the channel into STOPPED state. A START CHANNEL command must be issued to restart it.

CSQX498E

csect-name Invalid MQCD field *field-name*, value=*nnn* (X*xxx*)

Severity
8

Explanation

The MQCD structure returned by the channel auto-definition exit had an invalid value in the indicated field. The value is shown in decimal (*nnn*) and hexadecimal (*xxx*).

System action

The channel is not defined.

System programmer response

Correct the channel auto-definition exit.

CSQX500I

csect-name Channel *channel-name* started connection *conn-id*

Severity
0

Explanation

The specified channel has been started.

If *channel-name* is an inbound channel (indicated by *csect-name* containing CSQXRESP) then it was started from connection *conn-id*. If *channel-name* is an outbound channel then *conn-id* will be omitted.

System action

Processing continues.

CSQX501I

csect-name Channel *channel-name* no longer active connection *conn-id*

Severity
0

Explanation

Channel *channel-name* terminated. It is now inactive if it terminated normally when the disconnect interval expired, or stopped if it terminated because of an error or a STOP CHANNEL command.

If *channel-name* was an inbound channel (indicated by *csect-name* containing CSQXRESP) then it was started from connection *conn-id*. If *channel-name* was an outbound channel then *conn-id* will be omitted.

System action

Processing continues.

System programmer response

If the channel is stopped, resolve any error, and issue a START CHANNEL command to restart the channel.

CSQX502E

csect-name Action not allowed for channel *chl-type*(*channel-name*)

Severity
8

Explanation

The action you requested cannot be performed on channel *channel-name*. Some actions are only valid for certain channel types. This channel is a *chl-type* channel type. For example, you can only ping a channel from the end sending the message.

System action

The requested action is not performed.

System programmer response

Check whether the channel name is specified correctly. If it is, check that:

- The channel has been defined correctly
- The connection name identifies the remote end correctly
- For a cluster-receiver channel, the connection name does not specify a generic address or a shared listener port (INDISP=GROUP).
- For TCP/IP connections, the port number specified by the local channel matches that used by the listener at the remote queue manager.

You can use the *csect-name* to determine the action that failed:

Table 15. Mapping csect-names to actions	
csect-name	**action**
CSQXPING	PING CHANNEL
CSQXRESE	RESET CHANNEL
CSQXRESO	RESOLVE CHANNEL
CSQXSTOP	STOP CHANNEL

CSQX503E

csect-name Negotiation failed, channel *channel-name* type=*last-segment-type* data=*xxx* connection *conn-id*

Severity
8

Explanation

Channel *channel-name* could not be established due to a negotiation failure between the local queue manager and the remote end using connection *conn-id*. The last control data received was of type *last-segment-type* and is accompanied by data indicating the error.

A value of FFFFFFFF (-1) indicates that no error data was sent by the remote end.

System action

The channel is not started.

System programmer response

Examine the console log for the remote end for messages explaining the cause of the negotiation failure.

CSQX504E

csect-name Local protocol error, channel *channel-name* type=*type* data=*xxx*

Severity
8

Explanation

During communications with the remote end, the local message channel agent for channel *channel-name* detected a protocol error.

type shows the type of error that occurred and the incorrect value is shown by *xxx*.

00000001
Missing channel. Define a remote channel. See message CSQX520E for more information.

00000002
Incorrect channel type. Check your definitions. See message CSQX547E for more information.

00000003
Queue manager unavailable. Check the queue manager. See message CSQX524E for more information.

00000004
Message sequence error. Investigate the problem and reset the channel. See message CSQX526E for more information.

00000005
Queue manager terminating. This message might be for information only. See message CSQX525E for more information.

00000006
Unable to store. This message might be for information only. See messages CSQX527E and CSQX544E for more information. Also, check the error log for the remote system. Messages might end up on the remote dead-letter queue.

00000007
User closed. This message might be for information only. See message CSQX528I for more information. The channel is stopping, either because of a STOP CHANNEL command, or the channel initiator is stopping.

00000008
Timeout expired. This message might be for information only. During an MQGET_WAIT the DISCINT times out, so the channel is closed.

00000009
Target queue unknown - contact IBM Support.

0000000A
Incorrect segment type - contact IBM Support.

0000000B
Incorrect segment length. Check the remote client. Either the client has sent a segment larger than the buffer it requested, or the requested buffer exceeds the combined payload and header limits.

0000000C
Data not valid - contact IBM Support.

0000000D
Unexpected segment - contact IBM Support.

0000000E
Unexpected ID - contact IBM Support.

0000000F
Unexpected MSH - contact IBM Support.

00000010
General protocol problem - contact IBM Support.

00000011
Batch failure - contact IBM Support.

00000012
Incorrect message length - contact IBM Support.

00000013
Incorrect segment number - contact IBM Support.

00000014
Security failure - contact IBM Support.

00000015
Wrap value error. Use the command ALTER CHANNEL SEQWRAP to align the local or remote channel sequence wrap values. See message CSQX505E for more information.

00000016
Channel unavailable. Check if the remote channel is STOPPED, or otherwise unavailable. See message CSQX558E for more information.

00000017
Closed by exit - contact IBM Support.

00000018
Cipher spec error. Confirm the SSLCIPH of the channel, and its compatibility if the remote side has been set to SSLFIPS(YES). See message CSQX635E for more information.

00000019
Peer name error. Confirm that SSLPEERNAME on this channel, matches the distinguished name in the certificate of the remote side. See message CSQX636E for more information.

0000001A
SSL/TLS client certificate error. Check the remote channel and see if a certificate has been supplied for SSL/TLS negotiation. See message CSQX637E for more information.

0000001B
RMT RSRCS in recovery. This message is for information only; the condition is transient.

0000001C
SSL/TLS refreshing. This message is for information only; the condition is transient.

0000001D
HOBJ not valid - contact IBM Support.

0000001E
Conversion ID error - contact IBM Support.

0000001F
Socket action type not valid - contact IBM Support.

00000020
Standby queue manager not valid - contact your IBM support center.

00000021
Maximum transmission size not valid. Increase the remote RECEIVER attributes for transmission unit size.

00000022
FAP level not valid - contact IBM Support.

00000023
Maximum permitted conversions exceeded. The SHARECNV limit has been exceeded. Investigate the remote client and increase the value of SHARECNV.

00000024
Password protection error - contact IBM Support.

System action

The channel stops. The associated transmission queue might be set to GET(DISABLED) and triggering turned off.

System programmer response

Examine the console log to determine the cause of the failure. This might occur after the channel initiator or queue manager is stopped forcibly or ends abnormally. If it occurs in other cases, contact IBM Support to report the problem.

CSQX505E

csect-name Sequence wrap values differ, channel *channel-name* local=*local-seqno* remote=*remote-seqno*

Severity
8

Explanation

The sequence number wrap value for channel *channel-name* is *local-seqno*, but the value specified at the remote end is *remote-seqno*. The two values must be the same before the channel can be started.

System action

The channel does not start.

System programmer response

Change either the local or remote channel definition so that the values specified for the message sequence number wrap value are the same.

CSQX506E

csect-name Message receipt confirmation not received for channel *channel-name*

Severity
8

Explanation

The remote end did not accept the last batch of messages.

System action

Channel *channel-name* stops. The associated transmission queue may be set to GET(DISABLED) and triggering turned off.

System programmer response

Determine why the remote end did not accept the last batch of messages. Resolve the problem and restart the channel.

CSQX507E

csect-name Channel *channel-name* is in-doubt, connection *conn-id* (queue manager *qmgr-name*)

Severity
8

Explanation

Channel *channel-name* is in-doubt with the remote end using connection *conn-id*. The associated remote queue manager is *qmgr-name*; in some cases its name cannot be determined and so is shown as '????'.

System action

The requested operation does not complete.

System programmer response

Examine the status of the channel, and either restart a channel to resolve the in-doubt state, or use the RESOLVE CHANNEL command to correct the problem manually.

CSQX511I

csect-name Channel *channel-name* started, connection *conn-id*

Severity
0

Explanation

The specified SVRCONN channel has been started from connection *conn-id*.

System action

Processing continues.

CSQX512I

csect-name Channel *channel-name* no longer active, connection *conn-id*

Severity
0

Explanation

SVRCONN Channel *channel-name* terminated. It is now inactive if it terminated normally when the disconnect interval expired, or stopped if it terminated because of an error or a STOP CHANNEL command.

The SVRCONN *channel-name* was started from connection *conn-id*.

System action

Processing continues.

System programmer response

If the SVRCONN channel is stopped, resolve any error, and issue a START CHANNEL command to restart the channel.

CSQX513E

csect-name Current channel limit exceeded channel *channel-name* connection *conn-id*

Severity
8

Explanation

There are too many channels current to be able to start another. The maximum number allowed is specified in the MAXCHL queue manager attribute. Current channels include stopped and retrying channels as well as active channels.

If *channel-name* was an inbound channel (indicated by *csect-name* containing CSQXRESP) then it was started from connection *conn-id*. If *channel-name* was an outbound channel then *conn-id* will be omitted.

System action

The channel does not start.

System programmer response

Wait for some of the operating channels to terminate before restarting the channel, or use the ALTER QMGR command to increase **MAXCHL**. A change that increases **MAXCHL** will not be effective until the channel initiator has been stopped and restarted. If many of the currently operating channels are server-connection channels, consider limiting the number of those using **MAXINST** or **MAXINSTC** attributes of a server-connection channel. See Server-connection channel limits for more details.

CSQX514E

csect-name Channel *channel-name* is active on *qmgr-name*

Severity
8

Explanation

An operation was requested on a channel that is active. If the channel is shared, it might be active on another queue manager. If the channel is a receiver, a previous instance of it might have been orphaned and therefore still be active.

System action

The request fails.

System programmer response

For operations other than starting the channel, either stop the channel manually, or wait for it to terminate, and try the operation again. It might be necessary to use MODE(FORCE) to stop the channel manually if the Adopt MCA function is not being used. Using the Adopt MCA function avoids the need for manual intervention to handle orphaned receiver channels.

CSQX515I

csect-name Channel *channel-name* changed

Severity
0

Explanation

The channel for which information has been requested is a new instance of the channel. The previous channel instance has ended.

System action

The information shown is for the new channel instance.

CSQX516E

csect-name Error accessing synchronization data, RC=*return-code*

Severity
8

Explanation

There was an error when accessing the channel synchronization data.

If the return code is of the form 10009*nnn* or 20009*nnn*, it is a distributed queuing message code. This is generally associated with message CSQX*nnn*E, which will normally be issued previously.

Otherwise the most likely cause is a shortage of storage.

System action

The channel stops. The associated transmission queue may be set to GET(DISABLED) and triggering turned off.

In some cases, the channel initiator will stop as well.

System programmer response

If the return code is a distributed queuing message code, see the corresponding message explanation for more information. Where no such message is described, see "Distributed queuing message codes" on page 1147 for the corresponding message number.

Restart the channel or the channel initiator. If the problem persists, contact IBM Support.

CSQX517E

csect-name Error in *q-name* - channel *channel-name* repeated

Severity
8

Explanation

There was more than one set of synchronization information in *q-name* for an instance of channel *channel-name*. This is probably because the channel is a receiver channel, and there are two sender channels with the same name on different queue managers within the same network address that have communicated with it.

System action

The first set of synchronization information for the channel instance is used, and any others are ignored. Errors may occur if the channel is used.

System programmer response

Avoid using the channel. Remove the extra sets of information from the channel synchronization queue, and rename channels so that they have unique names.

If this does not resolve the problem, contact IBM Support.

CSQX519E

csect-name Channel *channel-name* not defined connection *remote-conn-id*

Severity
8

Explanation

The channel initiator could not find a definition of channel *channel-name*.

The associated remote connection name is *remote-conn-id*. If the request to use the channel is not from an inbound connection, or the remote connection name cannot be determined, *remote-conn-id* will be shown as '????'.

System action

The requested operation fails.

System programmer response

Ensure that the name is specified correctly and the channel definition is available.

The message can also be issued if an automatically defined cluster sender channel (CLUSSDRA) has been deleted as a result of issuing a REFRESH CLUSTER command and a putting application still has a queue object open which is using the channel.

CSQX520E

csect-name Remote channel *channel-name* not defined

Severity
8

Explanation

There is no definition of channel *channel-name* at the remote end.

System action

The channel does not start.

System programmer response

Add an appropriate channel definition at the remote end, and retry the operation.

CSQX523E

csect-name Remote protocol error, channel *channel-name* type=*type* data=*xxx*

Severity
8

Explanation

During communications with the remote end, the remote message channel agent for channel *channel-name* detected a protocol error. *type* shows the type of error that occurred:

0000000A
Incorrect segment type

0000000B
Incorrect length

0000000C
Invalid data

0000000D
Invalid segment

0000000E
Invalid ID

0000000F
Invalid MSH

00000010
General error

00000011
Batch failure

00000012
Incorrect message length

00000013
Incorrect segment number

The data associated with the error (for example, the incorrect value) is shown by *xxx*.

System action

The channel stops. The associated transmission queue might be set to GET(DISABLED) and triggering turned off.

System programmer response

Examine the console log for the remote end to determine the cause of the failure. This might occur after the channel initiator or queue manager is stopped forcibly or ends abnormally. If it occurs in other cases, contact IBM Support.

CSQX524E

csect-name Remote queue manager unavailable for channel *channel-name*

Severity
8

Explanation

Channel *channel-name* cannot start because the remote queue manager is not currently available.

System action

The channel does not start

System programmer response

Either start the remote queue manager, or retry the operation later.

CSQX525E

csect-name Channel *channel-name* closing because remote queue manager *qmgr-name* is stopping

Severity
8

Explanation

Channel *channel-name* is closing because the remote queue manager *qmgr-name* is stopping. In some cases, the remote queue manager name cannot be determined and so is shown as '????'.

System action

The channel stops. The associated transmission queue might be set to GET(DISABLED) and triggering turned off.

System programmer response

Investigate why the remote queue manager is stopping, if it was not expected.

CSQX526E

csect-name Message sequence error for channel *channel-name*, sent=*msg-seqno* expected=*exp-seqno*

Severity
8

Explanation

The local queue manager does not agree with the remote end on the next message sequence number for channel *channel-name*. The message is normally issued at both the sending and receiving end: at the sending end, *msg-seqno* and *exp-seqno* are unpredictable; at the receiving end, a message had sequence number *msg-seqno* but sequence number *exp-seqno* was expected.

System action

The channel stops. The associated transmission queue might be set to GET(DISABLED) and triggering turned off.

System programmer response

Determine the cause of the inconsistency. It could be that the synchronization information has become damaged, or has been backed out to a previous version. If the problem cannot be resolved, the sequence number can be reset manually at the sending end of the channel using the RESET CHANNEL command. (For some queue managers, it might be necessary to issue the RESET CHANNEL command at the receiving end as well.)

CSQX527E

csect-name Unable to send message for channel *channel-name*

Severity

8

Explanation

The remote end cannot receive the message that is being sent for channel *channel-name*.

System action

The channel stops. The associated transmission queue may be set to GET(DISABLED) and triggering turned off.

System programmer response

Examine the console log for the remote end to determine why the message cannot be received, and then restart the channel.

CSQX528I

csect-name Channel *channel-name* stopping

Severity

0

Explanation

The channel is closing because a STOP CHANNEL command was issued, or because the channel initiator is stopping.

System action

The channel stops. The associated transmission queue may be set to GET(DISABLED) and triggering turned off.

System programmer response

Note that a STOP CHANNEL command puts the channel into STOPPED state. A START CHANNEL command must be issued to restart it.

CSQX531E

csect-name Transmission queue *q-name* for *channel-name* has wrong usage type

Severity

8

Explanation

Queue *q-name* is named as a transmission queue in the channel definition for *channel-name*, but it is not a transmission queue.

System action

The channel does not start.

System programmer response

Ensure the queue name is specified correctly in the channel definition. If it is, alter the queue usage attribute of the queue to that of a transmission queue.

CSQX533I

csect-name Channel *channel-name* is already in requested state

Severity
0

Explanation

A request to stop channel *channel-name* was made, but the channel was already in the specified state, or in the process of reaching that state.

System action

The request is ignored.

CSQX534E

csect-name Channel *channel-name* is stopped

Severity
4

Explanation

The operation requested cannot be performed because the channel is currently stopped.

System action

The request is ignored.

System programmer response

Issue a START CHANNEL command to restart the channel.

CSQX535E

csect-name Channel *channel-name* stopping because exit *exit-name* is not valid

Severity
8

Explanation

The user exit *exit-name* specified for channel *channel-name* is not valid.

System action

The channel stops. The associated transmission queue might be set to GET(DISABLED) and triggering turned off. For auto-defined channels, the channel does not start.

System programmer response

Ensure that the user exit name is specified correctly in the channel definition, and that the user exit program is correct and available. The channel initiator loads exits from the library data sets under the CSQXLIB DD statement of its started task JCL procedure xxxxCHIN.

CSQX536I

csect-name Channel *channel-name* stopping because of request by exit *exit-name*

Severity
0

Explanation

The channel is closing because the user channel exit *exit-name* requested it.

System action

The channel stops. The associated transmission queue may be set to GET(DISABLED) and triggering turned off. For auto-defined channels, the channel does not start.

System programmer response

Note that this puts the channel into STOPPED state. A START CHANNEL command must be issued to restart it.

CSQX539E

csect-name Channel *channel-name* for queue *q-name* is not available

Severity
8

Explanation

A trigger message was received to start a channel *channel-name* to process the transmission queue *q-name*. However, the channel initiator could not find a defined and available channel to start.

System action

The channel does not start.

System programmer response

Ensure that there is a channel defined to process the transmission queue, and that it is not stopped.

CSQX540E

csect-name Unable to commit batch, channel *channel-name* MQCC=*mqcc* MQRC=*mqrc* (*mqrc-text*)

Severity
8

Explanation

An MQCMIT call for the queue associated with channel *channel-name* was unsuccessful.

System action

The channel stops. The associated transmission queue might be set to GET(DISABLED) and triggering turned off.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form).

CSQX541E

csect-name Invalid CCSIDs for data conversion, *ccsid1* and *ccsid2*

Severity
8

Explanation

Either the local coded character set identifier (CCSID) or the target CCSID is not valid, or is not currently supported, or conversion between the two CCSIDs involved is not supported. (The name of the channel cannot be determined because the invalid CCSID prevents the necessary data conversion being done.)

System action

The channel stops. The associated transmission queue may be set to GET(DISABLED) and triggering turned off.

System programmer response

Ensure that the CCSIDs are valid and that conversion between them is supported. For information about the CCSIDs that are supported, see Codeset names and CCSIDs.

CSQX544E

csect-name Messages for channel *channel-name* sent to remote dead-letter queue

Severity
4

Explanation

During the processing of channel *channel-name*, one or more messages have been put the dead-letter queue at the remote queue manager.

System action

Processing continues.

System programmer response

Examine the contents of the dead-letter queue. Each message is contained in a structure that describes why the message was put to the queue, and to where it was originally addressed.

CSQX545I

csect-name Channel *channel-name* closing because disconnect interval expired

Severity
0

Explanation

The channel is closing because no messages arrived on the transmission queue within the disconnect interval.

System action

The channel ends normally.

CSQX547E

csect-name Remote channel *channel-name* has the wrong type

Severity
8

Explanation

The operation requested cannot be performed because channel *channel-name* on the remote end is not of a suitable type. For example, if the local channel is defined as a sender the remote queue manager must define its corresponding channel as either a receiver or requester.

System action

The requested operation is not performed.

System programmer response

Check that the channel name is specified correctly. If it is, check that:

- The channel definition on the remote end has an appropriate channel type
- The connection name of the local channel identifies the remote end correctly
- For a cluster-receiver channel, the connection name does not specify a generic address or a shared listener port (INDISP=GROUP).
- For TCP/IP connections, the port number specified by the local channel matches that used by the listener at the remote queue manager.

CSQX548E

csect-name Messages sent to local dead-letter queue, channel *channel-name* reason=*mqrc* (*mqrc-text*)

Severity
4

Explanation

During the processing of channel *channel-name*, one or more messages have been put the dead-letter queue at the local queue manager. *mqrc* shows why, and is one of the following:

- an MQRC_* reason code from an MQPUT or MQPUT1 call
- an MQFB_* feedback code.

System action

Processing continues.

System programmer response

Examine the contents of the dead-letter queue. Each message is contained in a structure that describes why the message was put to the queue, and to where it was originally addressed.

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form).

For information about MQFB_* feedback codes see the MQMD description in MQMD - Message descriptor.

CSQX549E

csect-name Queue *q-name* for channel *channel-name* is get-inhibited

Severity
8

Explanation

An MQGET failed because the transmission queue had been previously inhibited for gets.

System action

The channel stops. The associated transmission queue might have triggering turned off.

System programmer response

Change the definition of the transmission queue so that it is not inhibited for MQGET calls.

CSQX551E

csect-name Action not supported, channel *channel-name* connection *conn-id* (queue manager *qmgr-name*)

Severity
8

Explanation

The operation requested for channel *channel-name* is not supported by the remote end using the connection *conn-id*. The associated remote queue manager is *qmgr-name*; in some cases its name cannot be determined and so is shown as '????'.

System action

The channel stops. The associated transmission queue may be set to GET(DISABLED) and triggering turned off.

System programmer response

Check that the connection name parameter is specified correctly and that the levels of the queue managers in use are compatible.

CSQX552E

csect-name Security exit data for channel *channel-name* not received, connection *conn-id*

Severity
8

Explanation

The local security user channel exit for channel *channel-name* requested data from the remote security user channel exit, but no data was received. The remote connection was *conn-id*.

System action

The channel stops. The associated transmission queue may be set to GET(DISABLED) and triggering turned off.

System programmer response

Ensure that the security exit for the channel on the remote end has been defined correctly and is available. If it is, check that the exit program operates correctly.

CSQX558E

csect-name Remote channel *channel-name* not available

Severity
8

Explanation

The channel *channel-name* at the remote end is currently stopped or is otherwise unavailable. For example, there might be too many channels current to be able to start it.

System action

The channel does not start.

System programmer response

This might be a temporary situation, and the channel will try again. If not, check the status of the channel at the remote end. If it is stopped, issue a START CHANNEL command to restart it. If there are too many channels current, either wait for some of the operating channels to terminate, or stop some channels manually, before restarting the channel.

CSQX565E

csect-name No dead-letter queue for *qmgr-name*, channel *channel-name*

Severity
8

Explanation

A message could not be delivered normally and there is no dead-letter queue defined for queue manager *qmgr-name*.

You can get this message with a cluster sender channel during message reallocation. During reallocation, the message is got from the transmission queue and put back again. If the transmission queue is full, then the put fails and tries writing the message to the dead letter queue. If the dead letter queue does not exist, message CSQX565E is produced, and the reallocation changes are rolled back. Reallocation does not happen until the queue full problem is resolved.

System action

The channel stops, except in the case where nonpersistent messages are being sent and the NPMCLASS attribute of the channel is set to FAST, when processing continues. The associated transmission queue may be set to GET(DISABLED) and triggering turned off.

System programmer response

Correct the problem that prevented the message from being delivered normally, or define a dead-letter queue for the remote queue manager.

CSQX567E

csect-name Listener unable to register to APPC/MVS, TRPTYPE=LU62 INDISP=*disposition* RC=*return-code* reason=*reason*

Severity

8

Explanation

While starting, the specified LU 6.2 listener could not register as an APPC/MVS server. The return code from APPC/MVS allocate services was *return-code* and the associated reason code was *reason* (both in hexadecimal).

System action

The listener is not started.

System programmer response

See "Communications protocol return codes for z/OS" on page 1132 for the cause of the return code from APPC/MVS allocate services, and the *z/OS MVS Programming: Writing Servers for APPC/MVS* manual for more information. Check that the LUNAME queue manager attribute is the same as the PARTNER_LU value for the APPC/MVS symbolic destination used by the listener.

CSQX568E

csect-name Listener unable to unregister from APPC/MVS, TRPTYPE=LU62 INDISP=*disposition* RC=*return-code* reason=*reason*

Severity

8

Explanation

While stopping, the specified LU 6.2 listener could not unregister as an APPC/MVS server. The return code from APPC/MVS allocate services was *return-code* and the associated reason code was *reason* (bothin hexadecimal).

System action

The listener stops. It may not be possible to restart it.

System programmer response

See "Communications protocol return codes for z/OS" on page 1132 for the cause of the return code from APPC/MVS allocate services and the *z/OS MVS Programming: Writing Servers for APPC/MVS* manual for more information.

CSQX569E

csect-name Channel *channel-name* exceeded TCP/IP channel limit

Severity

8

Explanation

The number of current TCP/IP channels is the maximum allowed; another channel cannot be started. Current channels include stopped and retrying channels as well as active channels. The maximum

allowed is specified in the TCPCHL queue manager attribute, but may be reduced if a dispatcher fails, or if TCP/IP resources are restricted (as reported by message CSQX118I).

System action

The channel does not start.

System programmer response

If the maximum allowed is zero, TCP/IP communications are not allowed, and no TCP/IP channels can be started. If the maximum allowed is non-zero, wait for some of the operating channels to terminate before restarting the channel, or use the ALTER QMGR command to increase TCPCHL.

CSQX570E

csect-name Channel *channel-name* exceeded LU 6.2 channel limit

Severity
8

Explanation

The number of current LU 6.2 channels is the maximum allowed; another channel cannot be started. Current channels include stopped and retrying channels as well as active channels. The maximum allowed is specified in the LU62CHL queue manager attribute, but may be reduced if a dispatcher fails.

System action

The channel does not start.

System programmer response

If the maximum allowed is zero, LU 6.2 communications are not allowed, and no LU 6.2 channels can be started. If the maximum allowed is non-zero, wait for some of the operating channels to terminate before restarting the channel, or use the ALTER QMGR command to increase LU62CHL.

CSQX571E

csect-name Error from PKCS #11 callable service '*func*', RC=*return-code*, reason=*reason*

Severity
8

Explanation

An attempt to use PKCS #11 callable service *func* failed.

System action

The component where the error occurred (message channel agent, supervisor) will continue but the feature being used will be unavailable.

If *func* is CSFPPRF (Pseudo-random function) the feature affected is password protection. If this feature is not being used then this error can be ignored. If this occurs at channel initiator startup, the password protection algorithm uses STCK instead.

System programmer response

For information about the *return-code* and *reason* from the PKCS #11 callable service, see the section on ICSF and cryptographic coprocessor return/reason codes in the *z/OS Cryptographic Services ICSF Application Programmer's Guide*.

For more information about Integrated Cryptographic Service Facility (ICSF), see Using ICSF.

CSQX572E

csect-name Channel *channel-name* stopping because message header is not valid

Severity
8

Explanation

During the processing of channel *channel-name*, a message was found that had an invalid header. The dead-letter queue was defined as a transmission queue, so a loop would have been created if the message had been put there.

System action

The channel stops. The associated transmission queue may be set to GET(DISABLED) and triggering turned off.

System programmer response

Correct the problem that caused the invalid message header.

CSQX573E

csect-name Channel *channel-name* exceeded active channel limit

Severity
8

Explanation

There are too many channels active (transmitting messages) to be able to start another. The maximum number allowed is specified in the ACTCHL queue manager attribute.

System action

The channel does not start.

System programmer response

Either wait for some of the operating channels to terminate, or stop some channels manually, before restarting the channel, or use the ALTER QMGR command to increase ACTCHL. A change that increases ACTCHL will not be effective until the channel inititator has been stopped and restarted.

CSQX574I

csect-name Channel *channel-name* can now start

Severity
0

Explanation

The specified channel was waiting to start, because there were too many channels active (transmitting messages) to be able to start another. One or more of the active channels has terminated, so this channel can now start.

Note: This message is not itself issued, although the corresponding event is generated.

CSQX575E

csect-name Negotiation failed for channel

Severity
8

Explanation

A channel between the local queue manager and the remote end could not be established due to a negotiation failure. The failure was such that the channel name could not be determined: for example, data conversion between the coded character set identifiers (CCSIDs) used by the local and remote ends might not have been possible.

System action

The channel is not started.

System programmer response

Examine the console log for the remote end for messages explaining the cause of the negotiation failure.

CSQX576E

csect-name ICSF is not available

Severity

8

Explanation

In order to generate entropy for the password protection algorithm, a call to CSFPPRF (Pseudo-random function) is made which requires the Integrated Cryptographic Service Facility (ICSF) to be available. ICSF was found not to be available.

System action

The password protection algorithm uses STCK instead.

System programmer response

If password protection is being used, start ICSF. If it is not being used, this error message can be ignored.

CSQX578E

csect-name Unable to save status for channel *channel-name*

Severity

8

Explanation

An internal error has occurred.

System action

The channel stops. The associated transmission queue may be set to GET(DISABLED) and triggering turned off.

Information about the error is written to the data set identified by the CSQSNAP DD statement of the channel initiator started task JCL procedure, xxxxCHIN.

System programmer response

Collect the items listed in the Problem Determination section and contact IBM Support.

CSQX599E

csect-name Channel *channel-name* ended abnormally connection *conn-id*

Severity

8

Explanation

Channel *channel-name* ended abnormally because of a severe problem, as reported in the preceding messages.

If *channel-name* is an inbound channel (indicated by *csect-name* containing CSQXRESP) then it was started from connection *conn-id*. If *channel-name* is an outbound channel then *conn-id* will be omitted. The *conn-id* may be followed by the resolved hostname or the network address in parentheses following the *conn-id* but this is dependent on whether it can be resolved and if there is sufficient space remaining to report it.

System action

The channel stops. The associated transmission queue might be set to GET(DISABLED) and triggering turned off.

System programmer response

Investigate the problem reported in the preceding messages. For more information see, Problem determination in DQM.

CSQX608E

csect-name Remote resources in recovery for channel *channel-name*

Severity
8

Explanation

Channel *channel-name* cannot start because resources at the remote queue manager are being recovered.

System action

The channel does not start.

System programmer response

Restart the channel at a later time. If the problem persists examine the console log for the remote end for messages explaining the cause of the problem. This includes an instance of CSQX609E with more details.

CSQX609E

csect-name Resources in recovery, channel *channel-name* MQCC=*mqcc* MQRC=*mqrc* (*mqrc-text*)

Severity
8

Explanation

The message channel agent for the channel could not connect to the queue manager because resources are being recovered.

System action

The channel does not start.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form), which come from an MQCONNX request.

CSQX613I

csect-name Channel *channel-name* instance is already in requested state

Severity
0

Explanation

A request to stop a particular instance of channel *channel-name* was made (by specifying a connection name or a remote queue manager name), but the channel instance was already in the specified state, or in the process of reaching that state.

This error will also apply if an attempt is made to stop a SVRCONN channel using the QMNAME parameter. In this case do not use the QMNAME parameter. In order to stop a specific SVRCONN instance use the CONNAME parameter

System action

The request is ignored.

CSQX616E

csect-name The proposed CipherSpec is not enabled. CipherSpec *cipherspec* channel *channel* connection *conn-id*

Severity
8

Explanation

A channel has failed to start because the other end has proposed a CipherSpec that is not enabled on the local channel initiator.

System action

The channel is prevented from starting.

System programmer response

Check that you have the correct digital certificate public key type for the CipherSpec you are trying to use; see Digital certificates and CipherSpec compatibility in IBM MQ for more information.

Examine the CipherSpec specified in the SSLCIPH channel attribute and consider using a more secure CipherSpec.

If the CipherSpec is for TLS 1.3 and TLS 1.3 is not enabled, enable TLS 1.3 by setting **AllowTLSV13**=*TRUE* in the TransportSecurity stanza in the QMINI data set in your queue manager start up procedure.

If you want to re-enable the use of weak or deprecated CipherSpecs, see Enabling deprecated CipherSpecs on z/OS.

Attention: Re-enabling CipherSpecs in this manner leaves systems exposed to possible security problems. You should use CipherSpecs that use only the TLS protocol, rather than SSLv3.

CSQX617I

csect-name SSL key repository refresh not processed, SSL communications unavailable

Severity
0

Explanation

The cached SSL key repository cannot be refreshed in response to a REFRESH SECURITY TYPE(SSL) command because SSL communications are currently unavailable.

System action

0

System programmer response

Investigate why SSL is not available and take action as appropriate. It may be necessary to restart the channel initiator to allow SSL to be used.

Note: Ensure that SSLTASKS is set to a nonzero value.

CSQX618I

csect-name SSL key repository refresh started

Severity
0

Explanation

The cached SSL key repository is being refreshed in response to a REFRESH SECURITY TYPE(SSL) command.

System action

Message CSQX619I will be issued when the refresh is complete.

CSQX619I

csect-name SSL key repository refresh processed

Severity
0

Explanation

The refresh of the cached SSL key repository is complete.

System action

Channels will be restarted as required.

CSQX620E

csect-name System SSL error, channel *channel-name* connection *conn-id* function '*func*' RC=*return-code*

Severity
8

Explanation

An unexpected SSL communications error occurred for a channel. The channel is *channel-name*; in some cases its name cannot be determined and so is shown as '????'. The remote connection is *conn-id*. *func* is the name of the System SSL function that gave the error, and *return-code* is the return code (in decimal unless *func* is 'gsk_fips_state_set' in which case it is in hexadecimal).

System action

The channel is stopped.

System programmer response

See “Transport Layer Security (TLS) return codes for z/OS” on page 1143 for the cause of the return code from System SSL and refer to SSL Function Return Codes in the *z/OS Cryptographic Services System SSL Programming* manual for more information.

CSQX625E

csect-name System SSL error, function '*func*' RC=*return-code*

Severity
8

Explanation

An unexpected SSL communications error occurred for an SSL server subtask. *func* is the name of the System SSL function that gave the error, and *return-code* is the return code (in decimal).

System action

The SSL server subtask terminates.

System programmer response

See “Transport Layer Security (TLS) return codes for z/OS” on page 1143 for the cause of the return code from System SSL and refer to SSL Function Return Codes in the *z/OS Cryptographic Services System SSL Programming* manual for more information.

CSQX629E

csect-name Channel *channel-name* requires ICSF for SSLCIPH(*ciph*)

Severity
8

Explanation

Channel *channel-name* is using a cipherspec *ciph* that requires Integrated Cryptographic Service Facility (ICSF) callable services, but ICSF is not available. Sometimes the channel name and cipherspec are unknown and so are shown as "????".

The 4-character hexadecimal codes are listed in Table 1 of *Enabling CipherSpecs* and Table 1 of *Deprecated CipherSpecs*.

If the CipherSpec is shown in the message as a 4-character hexadecimal code, the name can be obtained from CipherSpecs order in TLS handshake.

The cipherspecs that use GCM or ephemeral elliptic curve algorithms require ICSF.

System action

The channel will not start.

System programmer response

Ensure ICSF is available, or change the cipherspec that the channel is using to one that does not require ICSF. If you are using ICSF and running the queue manager with SSLFIPS(YES), ensure that ICSF is configured to run in FIPS mode.

For more information, see System SSL function return code 455 in the *z/OS Cryptographic Services System SSL Programming* manual.

CSQX630E

csect-name Channel *channel-name* requires SSL

Severity
8

Explanation

Channel *channel-name* cannot start because it requires SSL, but SSL communications are not currently available.

System action

The channel does not start.

System programmer response

If SSL is required, investigate why it is not available and take action as appropriate. One possible cause, is that there is no certificate available owned by the user who initiated the channel address space. If this is the case, you need to re-configure the user ID to have a certificate with the correct value, by issuing the command **RACDCERT ID(xxxx)**, where *xxxx* is the user ID.

Check that you have the SSL queue manager properties set, for example SSLTASKS must be greater than 0.

If SSL is not required, change the channel definition so that SSL is not used.

CSQX631E

csect-name Cipher specifications differ, channel *channel-name* local=*local-ciph* remote=*remote-ciph* connection *conn-id*

Severity
8

Explanation
The SSL cipher specification value for channel *channel-name* is *local-ciph*, but the value specified at the remote end (from connection *conn-id*) is *remote-ciph*.

Supported alias CipherSpecs are listed in CipherSpecs you can use with IBM MQ TLS support, and supported CipherSpecs and protocols are listed in CipherSpecs order in TLS handshake

System action

The channel does not start.

System programmer response

Change either the local or remote channel definition so that the values specified for the SSL cipher specification are the same.

If ANY_* alias CipherSpec values are being used, ensure that compatible CipherSpec values are used at both ends of the channel so that a suitable CipherSpec can be negotiated.

CSQX632I

csect-name SSL certificate has no associated user ID, remote channel *channel-name*, connection *conn-id* - channel initiator user ID used

Severity

0

Explanation

The certificate sent from the remote end (from connection *conn-id*) during SSL handshaking was accepted, but no user ID could be found associated with it. The channel is *channel-name*; in some cases its name cannot be determined and so is shown as '????'.

Likely causes are that the certificate or a matching certificate name filter are not defined to the external security manager (ESM), or that the certificate contains fields that are not understood by the ESM.

System action

The user ID of the channel initiator address space is used as the channel user ID for the channel.

System programmer response

If you are using certificate name filtering, you can create a filter that matches this certificate. See Working with Certificate Name Filters (CNFs) for details on associating a user ID with a certificate.

If the security you want on your channel does not require the use of the SSL mapped certificate user ID, you can define the channel to use Put Authority (**PUTAUT**) with a value of **ONLYMCA** instead of **DEF**, or **ALTMCA** instead of **CTX** and this message is not issued as no security checking for the channel is using the SSL mapped certificate user ID that could not be found. See Receiving channels using TCP/IP for more details about which user IDs are used for security checking on a receiving channel using TCP/IP.

Alternatively, change the **SSLPEER** channel attribute or create a **CHLAUTH** record to prevent this certificate being accepted from the remote channel. See Channel authentication records for more details.

CSQX633E

csect-name SSL certificate for remote channel *channel-name* failed local check, connection *conn-id*

Severity

8

Explanation

The certificate sent from the remote end (from connection *conn-id*) during SSL handshaking could not be validated. The channel is *channel-name*; in some cases its name cannot be determined and so is shown as '????'.

System action

The channel will not start.

System programmer response

Ensure that the SSL certificate connected to the key repository at the remote end is valid, and that the signing certificate(s) have been connected to the keyring on the local queue manager so that the certificate sent can be authenticated.

For full details about SSL certificates and key repositories see Securing.

This error might indicate that the remote end of the channel is configured to send the wrong certificate. Check the certificate label configuration at the remote end of the channel and ensure that the local key repository contains all of the necessary CA certificates.

Check if there is an expired certificate in the keyring that has the same **Subject Distinguished Name** as a non-expired certificate. If so, remove the expired one from the keyring.

For more information, refer to System SSL Function Return Code 8 in the *z/OS Cryptographic Services System SSL Programming* manual.

CSQX634E

csect-name SSL certificate failed remote check, channel *channel-name* connection *conn-id*

Severity
8

Explanation

The certificates sent to the remote end using the connection *conn-id* during SSL handshaking could not be validated. The channel is *channel-name*; in some cases its name cannot be determined and so is shown as '????'.

System action

The channel will not start.

System programmer response

Firstly, you need to check that the SSL certificate in the keyring at the local queue manager *qmgr-name* is valid, for example, in TRUST status and not expired.

Secondly, you also need to check that both the signing certificate (for example the certificate from the certificate authority) and the signed certificate have been connected to the key repository on the remote end, so that the certificate sent can be verified at the remote end.

The certificate used is either named on the channel in the CERTLABL attribute, or named on the queue manager in the CERTLABL attribute or CERTQSGL attribute (for a shared channel). If no certificate label is found in any of these attributes, then the certificate is named 'ibmWebSphereMQ*qsg-name*' (for a shared channel) or 'ibmWebSphereMQ*qmgr-name*', or a default certificate in the keyring is used.

For full details about SSL certificates and key repositories see Securing.

For more information, refer to System SSL Function return code 414 in the *z/OS Cryptographic Services System SSL Programming*manual.

CSQX635E

csect-name Invalid cipher specification *ciph* for channel *channel-name* connection *conn-id*

Severity
8

Explanation

The SSL cipher specification value for channel *channel-name* is not valid. The value is shown in the message as the full cipher string.

Supported CipherSpecs are listed in CipherSpecs order in TLS handshake.

This error can occur if the remote end is configured to use SSLFIPS(YES). Check the errors at the remote end to determine if this is the case.

This error can also occur if SSLFIPS is configured at either end and an attempt was made to start a channel with a TLS 1.3 CipherSpec. TLS 1.3 CipherSpecs have not been FIPS certified yet.

System action

The channel will not start.

System programmer response

Correct the SSL cipher specification for the channel. If the remote end is configured to only accept FIPS certified cipher specifications, change the channel to use a FIPS certified cipher spec. See Specifying CipherSpecs for details on which cipher specifications are FIPS certified.

For more information, refer to System SSL Function Return Code 402, System SSL Function Return Code 412, and System SSL Function Return Code 422 in the *z/OS Cryptographic Services System SSL Programming* manual.

If the CipherSpec is for TLS 1.3 and TLS 1.3 is not enabled, enable TLS 1.3 by setting **AllowTLSV13**=*TRUE* in the **TransportSecurity** stanza in the QMINI data set in your queue manager start up procedure. However, if the queue manager attribute SSLFIPS is set to YES, TLS V1.3 cipher specifications cannot be used because TLS V1.3 CipherSpecs are not yet FIPS certified. In this case, consider setting SSLFIPS to NO.

CSQX636E

csect-name Distinguished name does not match peer name, channel *channel-name* name='*dist-name*' connection *conn-id*

Severity
8

Explanation

The distinguished name, *dist-name*, specified in the SSL certificate at the remote end (from connection *conn-id*) does not match the SSL peer name for channel *channel-name*. The distinguished name at the remote end must match the peer name specified (which can be generic) before the channel can be started. In some cases the channel name cannot be determined and so is shown as '????'.

System action

The channel will not start.

System programmer response

This error might indicate that the remote end of the channel is configured to send the wrong certificate. Check the certificate label configuration at the remote end of the channel and ensure that the local key repository contains all of the necessary CA certificates.

To allow this remote end to connect, change the SSL peer name specification for the channel so that it matches the distinguished name in the SSL certificate at the remote end, or obtain the correct certificate for the remote end, as appropriate.

If the SSL Peer name specification needs to match a number of different distinguished names for multiple different remote SSL certificates, consider using channel authentication records to define rules to allow or block specific SSL peer names instead of the SSL Peer name specification on the channel definition. See Channel authentication records for more details.

CSQX637E

csect-name No SSL certificate for remote channel *channel-name*, connection *conn-id*

Severity
8

Explanation

The remote channel (from connection *conn-id*) did not supply a certificate to use during SSL handshaking, but a certificate is required. The channel is *channel-name*; in some cases its name cannot be determined and so is shown as '????'.

System action

The channel will not start.

System programmer response

Ensure that the SSL certificate is connected to the key repository of the remote end, and the certificate is marked as "TRUST" by RACF, and not expired. Alternatively, if appropriate, change the local channel definition so that its **SSLCAUTH** attribute is set to **OPTIONAL**.

For full details about SSL certificates and key repositories see Securing.

For more information, refer to System SSL Function Return Code 403 in the *z/OS Cryptographic Services System SSL Programming* manual.

CSQX638E

csect-name SSL communications error for channel *channel-name*, connection *conn-id*

Severity
8

Explanation

An unexpected SSL communications error occurred for a channel, as reported in the preceding messages. The channel is *channel-name*; in some cases its name cannot be determined and so is shown as '????'. The remote connection is *conn-id*.

System action

The channel will not start.

System programmer response

Investigate the problem reported in the preceding messages. Review the local and remote console logs for reports of network errors.

For more information, refer to System SSL Function Return Code 406 in the *z/OS Cryptographic Services System SSL Programming* manual.

CSQX639E

csect-name No cipher specification for remote channel *channel-name*, connection *conn-id*

Severity
8

Explanation

No SSL cipher specification was supplied by the remote channel *channel-name* (from connection *conn-id*), but one was required. In some cases the channel name cannot be determined and so is shown as '????'.

System action

The channel will not start.

System programmer response

Change the remote channel definition so that the value specified for the SSL cipher specification is the same as that of the local channel.

CSQX640E

csect-name Invalid peer name, channel *channel-name* attribute=*key-name*

Severity
8

Explanation

The SSL peer name for channel *channel-name* includes a distinguished name attribute key *key-name* which is invalid or unsupported. In some cases the channel name cannot be determined and so is shown as '????'.

System action

The channel will not start.

System programmer response

Correct the SSL peer name for the channel.

CSQX641E

csect-name Cipher specification error for remote channel, channel *channel-name*, local=*local-cipher*, remote=*remote-cipher*, connection *connection-id*

Severity
8

Explanation

An error occurred with the CipherSpec for remote channel *channel-name* (from connection *connection-id*). In some cases, the channel name cannot be determined and so is shown as '????'.

System action

The channel will not start.

System programmer response

Review the CipherSpecs at both ends of the channel and ensure that they are compatible.

If the CipherSpec is for TLS 1.3 and TLS 1.3 is not enabled, enable TLS 1.3 by setting **AllowTLSV13**=*TRUE* in the TransportSecurity stanza in the QMINI data set in your queue manager start up procedure.

If ANY_* alias CipherSpec values are being used, ensure that compatible CipherSpec values are used at both ends of the channel so that a suitable CipherSpec can be negotiated.

CSQX642E

csect-name No SSL certificate for channel *channel-name*

Severity
8

Explanation

The channel *channel-name* did not supply a certificate to use during SSL handshaking, but a certificate is required by the remote end. In some cases the channel name cannot be determined and so is shown as '????'.

System action

The channel does not start.

System programmer response

Ensure that the keyring of the local queue manager *qmgr-name* has an SSL certificate connected to it which is associated with the queue manager. If you have configured a certificate label, check that the certificate exists, is marked as "TRUST" by RACF, and not expired.

The certificate used is either named on the channel in the CERTLABL attribute, or named on the queue manager in the CERTLABL attribute or CERTQSGL attribute (for a shared channel). If no certificate label is found in any of these attributes, then the certificate is named 'ibmWebSphereMQ*qsg-name*' (for a shared channel) or 'ibmWebSphereMQ*qmgr-name*', or a default certificate in the keyring is used.

Alternatively, if appropriate, change the remote channel definition so that its SSLCAUTH attribute is set to OPTIONAL.

For full details about SSL certificates and key repositories, see Securing.

CSQX643E

csect-name Peer name error for remote channel *channel-name*, connection *conn-id*

Severity
8

Explanation

An error occurred with the SSL peer name for remote channel *channel-name* (from connection *conn-id*). In some cases the channel name cannot be determined and so is shown as '????'.

System action

The channel will not start.

System programmer response

Review the remote console log to determine the peer name error.

CSQX644E

csect-name Unable to determine peer name for remote channel *channel-name*

Severity
4

Explanation

The peer name associated with the certificate sent from the remote end during SSL handshaking could not be determined. The channel is *channel-name*; in some cases its name cannot be determined and so is shown as '????'.

System action

If the local channel has a peer name specified it does not start.

System programmer response

Ensure that the SSL certificate in the keyring at the local queue manager *qmgr-name* is valid, and that the signing certificate has been connected to the key repository on the remote end so that the certificate sent can be authenticated.

The certificate used is either named on the channel in the CERTLABL attribute, or named on the queue manager in the CERTLABL attribute or CERTQSGL attribute (for a shared channel). If no certificate label is found in any of these attributes, then the certificate is named 'ibmWebSphereMQ*qsg-name*' (for a shared channel) or 'ibmWebSphereMQ*qmgr-name*', or a default certificate in the keyring is used.

Check that the local and remote channel definitions are correct.

For full details about SSL certificates and key repositories, see Securing.

CSQX645E

csect-name Certificate *cert-label* missing for channel *channel-name*

Severity
4

Explanation

An SSL/TLS certificate *cert-label*, or the default certificate cannot be found in the keyring or the certificate is not trusted. The channel is *channel-name*; in some cases its name cannot be determined and so is shown as '????'.

In some cases this message will appear multiple times, once for each affected channel.

System action

The channel does not start.

System programmer response

Ensure that the SSL/TLS certificate named *cert-label* is in the keyring and that it is valid.

Alternatively, change the certificate label configuration so that the channel uses a valid certificate.

The certificate used is either named on the channel in the CERTLABL attribute, or named on the queue manager in the CERTLABL attribute or CERTQSGL attribute (for a shared channel). If no certificate label is found in any of these attributes, then the certificate is named 'ibmWebSphereMQ*qsg-name*' (for a shared channel) or 'ibmWebSphereMQ*qmgr-name*', or a default certificate in the keyring is used.

To verify which keyring is in use, issue the following MQSC command:

```
DISPLAY QMGR SSLKEYR
```

To list the certificates that are present in the keyring in use, issue the following RACF command, or an equivalent command in your External Security Manager:

```
RACDCERT ID(chinit-user-id) LISTRING(key-ring-name)
```

For more information, refer to return codes System SSL Function Return Code 6 and System SSL Function Return Code 407 in the *z/OS Cryptographic Services System SSL Programming* manual.

CSQX646E

csect-name Error accessing LDAP server for channel *channel-name*

Severity
4

Explanation

While checking CRLs for a channel, an error occurred in setting up the LDAP environment or retrieving an LDAP directory entry. The channel is *channel-name*; in some cases its name cannot be determined and so is shown as '????'.

System action

The channel will not start.

System programmer response

Ensure that the LDAP server is specified and set up correctly, and is running.

For more information, refer to System SSL Function Return Code 11in the *z/OS Cryptographic Services System SSL Programming* manual.

CSQX658E

csect-name SSL certificate has expired, channel *channel-name* connection *conn-id*

Severity
4

Explanation

The current time is either before the SSL certificate start time or or after the end time. The channel is *channel-name*; in some cases its name cannot be determined and so is shown as '????'. The connection is *conn-id*.

System action

The channel will not start.

System Programmer response

Obtain a new certificate if the certificate has expired, or wait until the certificate becomes valid if it is not valid yet.

For more information, refer to System SSL Function Return Code 401 in the *z/OS Cryptographic Services System SSL Programming* manual.

CSQX663E

csect-name SSL certificate signature is incorrect, channel *channel-name* connection *conn-id*

Severity
4

Explanation

In the SSL certificate sent from the remote end using the connection *conn-id*, the certificate signature is not correct. The channel is *channel-name*; in some cases its name cannot be determined and so is shown as '????'.

System action

The channel will not start.

System programmer response

Ensure that the SSL certificate connected to the key repository at the remote end is valid.

For more information, refer to System SSL Function Return Code 413 in the *z/OS Cryptographic Services System SSL Programming* manual.

CSQX665E

csect-name Channel *channel-name* stopping because remote SSL socket closed, connection *conn-id*

Severity
4

Explanation

The remote end of a channel using SSL communications (from connection *conn-id*) closed the socket or sent a close notification alert. The channel is *channel-name*; in some cases its name cannot be determined and so is shown as '????'.

System action

The channel stops.

System programmer response

Examine the console log for the remote end to determine the cause of the failure.

For more information, refer to System SSL Function Return Code 420 in the *z/OS Cryptographic Services System SSL Programming* manual.

CSQX666E

csect-name LDAP server unavailable for channel *channel-name*

Severity
4

Explanation

While checking CRLs for a channel, the required LDAP server was not available. The channel is *channel-name*; in some cases its name cannot be determined and so is shown as '????'.

System action

The channel does not start.

System programmer response

Ensure that the LDAP server is running.

For more information, refer to System SSL Function return Code 427 in the *z/OS Cryptographic Services System SSL Programming* manual.

CSQX668I

csect-name Cipher specifications based on the TLS V1.2 protocol are disabled

Severity
4

Explanation

Cipher specifications that use the TLS V1.2 protocol are not enabled. Channels configured to use those cipher specifications fail when started.

System action

Processing continues.

System programmer response

If you do not need to use cipher specifications based on the TLS V1.2 protocol, no action is required.

If you want to enable the use of cipher specifications based on the TLS V1.2 protocol, either stop the queue manager and remove or comment out the list of AllowedCipherSpecs defined in the QMINI dataset at queue manager start up, or add cipher specifications based on the TLS V1.2 protocol to the list of AllowedCipherSpecs. Restart the queue manager.

CSQX669I

csect-name Cipher specifications based on the TLS V1.2 protocol are enabled

Severity
4

Explanation

Cipher specifications based on the TLS V1.2 protocol are enabled, and channels can be configured to use those cipher specifications.

System action

Processing continues.

System programmer response

If you do not need to use cipher specifications based on the TLS V1.2 protocol, stop the queue manager and define an AllowedCipherSpecs list in the QMINI data that does not contain any cipher specifications based on the TLS V1.2 protocol. Restart the queue manager.

CSQX670I

csect-name Cipher specifications based on the TLS V1.3 protocol are disabled

Severity
4

Explanation

Cipher specifications that use the TLS V1.3 protocol are not enabled. Channels configured to use those cipher specifications fail when started.

System action

Processing continues.

System programmer response

If you do not need to use cipher specifications based on the TLS V1.3 protocol, no action is required.

If you want to enable cipher specifications based on the TLS V1.3 protocol, confirm that the queue manager is running z/OS 2.4 or later. If not the version of z/OS will need to be upgraded.

If the queue manager is already running on a version of the z/OS operating system that supports TLS 1.3, stop the queue manager and set the **AllowTLSV13** property in the **TransportSecurity** stanza in the QMINI dataset to *TRUE*. By default, TLS 1.3 is disabled on queue managers that have been migrated to IBM MQ 9.2.0 or later and enabled on queue managers that have been newly created to run at IBM MQ 9.2.0 or later. Also, if you have defined an AllowedCipherSpecs list in the QMINI dataset, add cipher specifications based on the TLS 1.3 protocol to the list. Restart the queue manager.

If the queue manager attribute SSLFIPS is set to YES, TLS 1.3 cipher specifications cannot be used because TLS 1.3 CipherSpecs are not yet FIPS certified. Consider setting SSLFIPS to NO.

If the queue manager attribute **SSLKEYR** contains a blank value, ensure that **SSLKEYR** references the defined keyring.

CSQX671I

csect-name Cipher specifications based on the TLS V1.3 protocol are enabled

Severity
4

Explanation

Cipher specifications based on the TLS V1.3 protocol are enabled, and channels can be configured to use those cipher specifications.

System action

Processing continues.

System programmer response

None, unless you want to use a cipher specification that has been disabled because of TLS 1.3 being enabled. If so, see Note 3 in Deprecated CipherSpecs.

CSQX673E

csect-name Certificate label *cert-label* not used on channel *channel-name*, remote connection *conn-id*

Severity
8

Explanation

The SSL or TLS channel *channel-name* is configured to use certificate label *cert-label*. However, the remote peer did not send the necessary information to allow the local channel to use the correct certificate. The remote host is *conn-id*.

This error occurs when the local channel definition has a certificate label and the remote peer does not support selection of certificates.

System action

The channel will not start.

System programmer response

Ensure that the remote peer supports certificate label configuration. Refer to Digital certificate labels, understanding the requirements for details of certificate label requirements. Alternatively, alter the local channel definition so that it does not specify a certificate label.

CSQX674E

csect-name Channel *channel-name* specified a weak or broken SSL CipherSpec *sslciph*

Severity
8

Explanation

The channel is unable to start because it is configured to use a CipherSpec that is potentially insecure.

System action

The channel is prevented from starting.

System programmer response

Examine the CipherSpec specified in the SSLCIPH channel attribute and consider using a more secure CipherSpec.

If the CipherSpec is for TLS 1.3 and TLS 1.3 is not enabled, enable TLS 1.3 by setting **AllowTLSV13**=*TRUE* in the TransportSecurity stanza in the QMINI data set in your queue manager start up procedure.

If you want to re-enable the use of weak or deprecated CipherSpecs, see Enabling deprecated CipherSpecs on z/OS.

Attention: Re-enabling CipherSpecs in this manner leaves systems exposed to possible security problems. You should use CipherSpecs that use only the TLS protocol, rather than SSLv3.

CSQX675E

csect-name Unable to complete SSL key repository refresh

Severity

4

Explanation

The refresh of the cached SSL key repository could not be completed because of errors.

System action

The refresh is incomplete.

System programmer response

Examine the console log for messages that might indicate why the refresh could not be started.

CSQX676E

csect-name SSL key repository refresh completed, but some channels not restarted

Severity

4

Explanation

The refresh of the cached SSL key repository has completed, so the latest values and certificates are in use for all SSL channels. However, not all the outbound SSL channels which were running when the refresh was initiated could be restarted after the refresh had completed.

System action

Processing continues.

System programmer response

Examine the console log for messages identifying the channels that did not restart.

CSQX677E

csect-name SSL key repository refresh terminated, waiting for channel *channel-name*

Severity

4

Explanation

The cached SSL key repository is being refreshed, which involves stopping all the channels that use SSL communications. One or more of the channels is taking too long to stop. The channel is *channel-name*; in some cases its name cannot be determined and so is shown as '????'.

System action

The refresh is terminated. Some channels using SSL will have been stopped.

System programmer response

Stop any SSL channels that have not already stopped and issue the REFRESH SECURITY TYPE(SSL) command again.

CSQX678E

csect-name Channel *channel-name* not started, refreshing SSL key repository

Severity
4

Explanation

A channel using SSL communications could not be started because the cached SSL key repository is currently being refreshed. The channel is *channel-name*; in some cases its name cannot be determined and so is shown as '????'.

System action

The channel does not start.

System programmer response

Wait until the refresh has completed and start the channel again.

CSQX679E

csect-name Channel *channel-name* not started, refreshing remote SSL key repository

Severity
4

Explanation

A channel using SSL communications could not be started because the cached SSL key repository is currently being refreshed at the remote end. The channel is *channel-name*; in some cases its name cannot be determined and so is shown as '????'.

System action

The channel does not start.

System programmer response

Wait until the refresh has completed and start the channel again.

CSQX683E

csect-name SSL key repository has no certificates

Severity
4

Explanation

The SSL key repository (that is, the keyring in the external security manager) does not contain any valid certificates.

System action

Channels using SSL communications will not start.

System programmer response

Add the user certificate and any necessary certificate authority (CA) certificates to the key repository. Ensure that existing certificates are valid, have not expired, and are marked as trusted.

For more information, refer to System SSL Function Return Code 7 in the *z/OS Cryptographic Services System SSL Programming* manual.

CSQX684E

csect-name SSL key repository has no CA certificates

Severity
4

Explanation

The SSL key repository (that is, the keyring in the external security manager) does not contain any valid certificate authority (CA) certificates. A channel using SSL communications needs at least one CA or self-signed certificate to perform client authentication.

System action

Channels using SSL communications will not start.

System programmer response

Add the user certificate and any necessary certificate authority (CA) certificates to the key repository. Ensure that existing certificates are valid, have not expired, and are marked as trusted.

For more information, refer to System SSL Function Return Code 109 in the *z/OS Cryptographic Services System SSL Programming* manual.

CSQX685E

csect-name No self-signed certificate for channel *channel-name*, connection *conn-id*

Severity
4

Explanation

A self-signed certificate cannot be validated as it is not in the SSL key repository. The channel is *channel-name*; in some cases its name cannot be determined and so is shown as '????'. The remote connection is *conn-id*.

System action

The channel is not started.

System programmer response

Add the self-signed certificate to the key repository.

Note: Changes to the key repository do not take effect immediately, see When changes to certificates or the key repository become effective on z/OS. If you have already added the self-signed certificate to the key repository, issue a REFRESH SECURITY TYPE(SSL) command or recycle the CHINIT address space.

For more information, refer to System SSL Function Return Code 417 in the *z/OS Cryptographic Services System SSL Programming* manual.

CSQX686E

csect-name SSL private key error for channel *channel-name*

Severity
4

Explanation

The SSL certificate used has no associated private key, or the private key is not available because it key is stored in ICSF and ICSF services are not available. The channel is *channel-name*; in some cases its name cannot be determined and so is shown as '????'.

The certificate used is either named on the channel in the CERTLABL attribute, or named on the queue manager in the CERTLABL attribute or CERTQSGL attribute (for a shared channel). If no certificate label is found in any of these attributes, then the certificate is named 'ibmWebSphereMQ*qsg-name*' (for a shared channel) or 'ibmWebSphereMQ*qmgr-name*', or a default certificate in the keyring is used.

System action

The channel is not started.

System programmer response

Ensure that the private key associated with the SSL certificate used is available. Ensure that the ICSF started task is running if the private key is stored in ICSF. See Giving the channel initiator the correct access rights on z/OS for information on how to actually give the CHINIT the proper accesses to get to its keys (if in ICSF).

For more information, refer to System SSL Function Return Code 428 in the *z/OS Cryptographic Services System SSL Programming* manual.

CSQX687E

csect-name SSL certificate revoked by CA for channel *channel-name*, connection *conn-id*

Severity
4

Explanation

The SSL certificate has been revoked by the certificate authority (CA). The channel is *channel-name*; in some cases its name cannot be determined and so is shown as '????'. The remote connection is *conn-id*.

System action

The channel is not started.

System programmer response

Obtain a new certificate and add it to the key repository.

For more information, refer to System SSL Function Return Code 431 in the *z/OS Cryptographic Services System SSL Programming* manual.

CSQX688E

csect-name No SSL CA certificate for channel *channel-name*, connection *conn-id*

Severity
4

Explanation

The SSL key repository does not contain a certificate for the certificate authority (CA). The channel is *channel-name*; in some cases its name cannot be determined and so is shown as '????'. The remote connection is *conn-id*.

System action

The channel is not started.

System programmer response

Obtain a certificate for the certificate authority (CA) and add it to the key repository.

For more information, refer to System SSL Function Return Code 435 in the *z/OS Cryptographic Services System SSL Programming* manual.

CSQX689E

csect-name CRL cannot be processed for channel *channel-name*, connection *conn-id*

Severity
4

Explanation

A Certificate Revocation List (CRL) is not valid and cannot be processed. The channel is *channel-name*; in some cases its name cannot be determined and so is shown as '????'. The remote connection is *conn-id*.

System action

The channel is not started.

System programmer response

Contact the certificate authority and obtain a replacement CRL.

For more information, refer to System SSL Function Return Code 436 in the *z/OS Cryptographic Services System SSL Programming* manual.

CSQX690I

csect-name Cipher specifications based on the SSLv3 protocol are disabled.

Severity
4

Explanation

Cipher specifications based on the SSLv3 protocol are not enabled, and channels configured to use those cipher specifications fail when started.

System action

Processing continues.

System programmer response

If you do not need to use cipher specifications based on the SSLv3 protocol, then no action is required.

If you want to re-enable the use of weak or deprecated CipherSpecs, see Enabling deprecated CipherSpecs on z/OS.

By default, cipher specifications based on the SSLv3 protocol are disabled when the TLS V1.3 protocol is enabled. If you really do need to use cipher specifications based on SSLv3, consider disabling the TLS V1.3 protocol. Refer to message CSQX671I, although you should enable the stronger TLS V1.3 protocol.

Attention: Re-enabling CipherSpecs in this manner leaves systems exposed to possible security problems. You should use CipherSpecs that utilize only the TLS V1.3 or V1.2 protocols, rather than the SSLv3 protocol.

CSQX691I

csect-name Cipher specifications based on the SSLv3 protocol are enabled.

Severity
4

Explanation

Cipher specifications based on the SSLv3 protocol are enabled, and channels can be configured to use those cipher specifications.

System action

Processing continues.

System programmer response

If you need to use cipher specifications based on the SSLv3 protocol, then no action is required.

If you do not need to use cipher specifications based on the SSLv3 protocol, you should remove the override that enables the use of SSLv3.

See message CSQX690I for information on enabling SSLv3.

CSQX692I

csect-name Weak or broken SSL cipher specifications are disabled.

Severity
4

Explanation

Cipher specifications that are known to be weak or broken are not enabled, This includes all SSLv3-based cipher specifications. Channels configured to use those cipher specifications fail when started.

System action

Processing continues.

System programmer response

If you do not need to use broken or weak cipher specifications, no action is required.

If you want to re-enable the use of weak or deprecated CipherSpecs, see Enabling deprecated CipherSpecs on z/OS.

Attention: Re-enabling CipherSpecs in this manner leaves systems exposed to possible security problems. You should use CipherSpecs that utilize only the TLS V1.3 or V1.2 protocols, rather than the SSLv3 protocol.

CSQX693I

csect-name Weak or broken SSL cipher specifications are enabled.

Severity
4

Explanation

Cipher specifications known to be weak or broken are enabled, and channels can be configured to use those cipher specifications.

System action

Processing continues.

System programmer response

If you need to use weak or broken cipher specifications, no action is required.

If you do not need to use weak or broken cipher specifications, you should remove the override that enables the use of weak or broken cipher specifications.

See message CSQX692I for information on enabling weak or broken cipher specifications.

CSQX694I

csect-name Cipher specifications based on the TLS V1.0 protocol are disabled.

Severity
4

Explanation

Cipher specifications that use the TLS V1.0 protocol are not enabled. Channels configured to use those cipher specifications fail when started.

System action

Processing continues.

System programmer response

If you do not need to use cipher specifications based on the TLS V1.0 protocol, no action is required.

If you want to re-enable the use of weak or deprecated CipherSpecs, see Enabling deprecated CipherSpecs on z/OS.

By default, weak or deprecated cipher specifications are disabled when protocol TLS V1.3 is enabled. If you really do need to use cipher specifications based on SSLv3, consider disabling the TLS V1.3 protocol. Refer to message CSQX671I, although you should enable the stronger TLS V1.3 protocol.

Attention: Re-enabling CipherSpecs in this manner leaves systems exposed to possible security problems. You should use CipherSpecs that use only the TLS V1.3 or TLS V1.2 protocols, rather than the SSLv3 protocol.

CSQX695I

csect-name Cipher specifications based on the TLS V1.0 protocol are enabled.

Severity
4

Explanation

Cipher specifications based on the TLS V1.0 protocol are enabled, and channels can be configured to use those cipher specifications.

System action

Processing continues.

System programmer response

If you need to use weak or broken cipher specifications, no action is required.

If you do not need to use cipher specifications based on the TLS V1.0 protocol, you should remove the override that enables the use of TLS 1.0 cipher specifications.

See message CSQX694I for information on enabling cipher specifications based on TLS V1.0.

CSQX697I

csect-name Listener will only negotiate System SSL default cipher specifications.

Severity
4

Explanation

The listener will only negotiate with cipher specifications that are listed by default on the **System SSL** default cipher specification list.

System action

Processing continues.

System programmer response

If you only want to be able to negotiate with the listener using the cipher specifications listed on the **System SSL** default cipher specification list, then you can enable this behavior by adding a dummy Data Definition (DD) statement named **GSKDCIPS** to the channel initiator JCL; see Enabling deprecated CipherSpecs on z/OS.

CSQX703W

csect-name Message sequence number mismatch detected for channel *channel-name*

Severity
4

Explanation
The local and remote queue managers do not agree on the next message sequence number. A message with sequence number <insert_1> has been sent when sequence number <insert_2> was expected. The remote host is <insert_4>. The channel will ignore the mismatch in sequence numbers and attempt to continue.

System action
The sequence number mismatch is ignored, and the channel continues startup.

System programmer response
Determine the cause of the inconsistency. It could be that a channel has been deleted and redefined, an MCA has previously ended abnormally whilst confirming a message batch, or the channel is being intentionally redirected to an equivalent queue manager (that is, disaster recovery).

CSQX705E

csect-name Remote channel exit load error.

Severity
8

Explanation

A channel exit specified on the remote host could not be loaded.

System action

The channel ends abnormally .

System programmer response

Check the error logs on the remote host and ensure that the user exit is specified correctly in the channel definition.

CSQX769I

Authentication failed for channel *channel-name* from *ip-address*, unknown user *user-id*

Severity
4

Explanation

The authentication for a client connection has failed due to the user not being defined to the security manager (for example, RACF).

System action

The channel is not started.

System programmer response

Ensure that the application provides a user id that is registered with the security manager, or change the queue manager connection authority (CONNAUTH) configuration.

CSQX772E

csect-name mqapi-call failed, MQRC=*mqrc* (*mqrc-text*)

Severity
8

Explanation

The indicated IBM MQ *mqapi-call* failed for the specified reason code *mqrc*, (*mqrc-text*).

System action

Typically the component in which the error occurs terminates. When the component is a message channel agent, the associated channel is stopped.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqrc* (*mqrc-text* provides the MQRC in textual form).

CSQX774E

csect-name CHLAUTH cache load failed, all inbound channels blocked

Severity
8

Explanation

The CHLAUTH cache has failed to load. All inbound channels has been blocked from starting until the problem has been fixed. See previous message for the cause of the problem.

System action

All inbound channels are blocked from starting.

System programmer response

Look for the previous related message for the cause of the problem.

CSQX775I

csect-name Channel *channel-name* from *ipaddress* would have been blocked due to userid, Detail: *detail*

Severity
4

Explanation

The inbound channel *channel-name* would have been blocked from address *ipaddress* because the active values of the channel were mapped to a userid that should be blocked. Access is allowed as the channel authentication record is in warning mode.

The active values of the channel were *detail*.

System action

The channel is started.

System programmer response

Examine the channel authentication records to ensure that the correct settings have been configured. If the channel authentication record was not in warning mode the channel would be blocked. The ALTER QMGR **CHLAUTH** switch is used to control whether the channel authentication records are used. The DISPLAY CHLAUTH command can be used to query the channel authentication records.

CSQX776E

csect-name Channel *channel-name* from *ipaddress* has been blocked due to userid, Detail: *detail*

Severity
8

Explanation

The inbound channel *channel-name* was blocked from address *ipaddress* because the active values of the channel were mapped to a userid that should be blocked.

The active values of the channel were *detail*.

System action

The channel is not started.

System programmer response

Examine the channel authentication records to ensure that the correct settings have been configured. The ALTER QMGR **CHLAUTH** switch is used to control whether the channel authentication records are used. The DISPLAY CHLAUTH command can be used to query the channel authentication records.

CSQX777E

csect-name Channel *channel-name* from *ipaddress* has been blocked due to USERSRC(NOACCESS), Detail: *detail*

Severity
8

Explanation

The inbound channel *channel-name* was blocked from address *ipaddress* because the active values of the channel matched a channel authentication record configured with USERSRC(NOACCESS).

The active values of the channel were *detail*.

System action

The channel is not started.

System programmer response

Examine the channel authentication records to ensure that the correct settings have been configured.

The ALTER QMGR **CHLAUTH** switch is used to control whether the channel authentication records are used. The DISPLAY CHLAUTH can be used to query the channel authentication records.

If no host name is shown in the message next to the IP address, and CHLAUTH rules using host names are in place, ensure that your Domain Name Servers can correctly resolve the IP address to a host name and that your queue manager is configured with REVDNS(ENABLED).

CSQX782E

csect-name Connection from address *ipaddress* has been blocked due to matching rule *ip-address-pattern*

Severity
8

Explanation

The inbound connection from the address was blocked because it matches one of the blocked addresses, *ip-address-pattern*, in the channel authentication table.

System action

The channel is not started.

System programmer response

Examine the channel authentication records to ensure that the correct settings have been configured. The ALTER QMGR **CHLAUTH** switch is used to control whether the channel authentication records are used. The DISPLAY CHLAUTH can be used to query the channel authentication records.

CSQX785E

csect-name Channel *channel-name* is configured to not use the dead-letter queue

Severity
8

Explanation

Channel *channel-name* failed to deliver a message to its destination. The report option MQRO_DISCARD_MSG was not specified for the message and the channel has been configured to not use the dead-letter queue through the attribute setting USEDLQ(NO).

System action

The channel either discards the message, or the channel ends, in accordance with the NPMSPEED attribute setting.

System programmer response

Investigate the cause of this error, then either correct the problem that prevented the channel delivering the message, or enable the channel to use the dead-letter queue.

CSQX786I

csect-name Connection from address *ipaddress* would have been blocked due to matching rule *ip-address-pattern*

Severity
4

Explanation

The inbound connection from the address *ipaddress* would have been blocked because it matches one of the blocked addresses, *ip-address-pattern*, in the channel authentication table. Access is allowed as the channel authentication table is in warning mode.

System action

The channel is started.

System programmer response

Examine the channel authentication records to ensure that the correct settings have been configured. If the channel authentication record was not in warning mode the channel would be blocked. The ALTER QMGR **CHLAUTH** switch is used to control whether the channel authentication records are used. The DISPLAY CHLAUTH command can be used to query the channel authentication records.

CSQX787I

csect-name Channel *channel-name* from *ipaddress* would have been blocked due to USERSRC(NOACCESS), Detail: *detail*

Severity
4

Explanation

The inbound channel *channel-name* would have been blocked from address *ipaddress* because the active values of the channel matched a channel authentication record configured with USERSRC(NOACCESS). It was not blocked due to the channel authentication record being in warning mode.

The active values of the channel were *detail*.

System action

The channel is started.

System programmer response

Examine the channel authentication records to ensure that the correct settings have been configured. If the channel authentication record was not in warning mode the channel would be blocked. The ALTER QMGR **CHLAUTH** switch is used to control whether the channel authentication records are used. The DISPLAY CHLAUTH command can be used to query the channel authentication records.

CSQX788I

csect-name DNS lookup for address *address* using function '*func*' took *n* seconds

Severity
4

Explanation

An attempt to resolve address *address* using the '*func*' function call took *n* seconds to complete. This might indicate a problem with the DNS configuration.

System action

Processing continues.

System programmer response

Ensure that the DNS is correctly configured on the local system.

If the address was an IP address then the slow operation was a reverse DNS lookup. Some DNS configurations are not capable of reverse DNS lookups and some IP addresses have no valid reverse DNS entries.

If the problem persists, consider disabling reverse DNS lookups until the issue with the DNS can be resolved.

CSQX790I

csect-name Connection authentication failed for user *user-id* due to CHLAUTH with CHCKCLNT(*chckclnt-value*), Detail: *detail*

Severity
4

Explanation

The user ID *user-id* and its password were checked because the inbound connection matched a channel authentication record with CHCKCLNT(*chckclnt-value*).

The active values of the channel were *detail*. The MATCH(RUNCHECK) mode of the DISPLAY CHLAUTH command can be used to identify the relevant CHLAUTH record.

This message accompanies a previous error to clarify the reason for the user ID and password check.

System action

The channel is not started.

System programmer response

Refer to the previous error for more information.

Ensure that a password is specified by the client application and that the password is correct for the User ID.

Alternatively, to avoid the authentication check you can amend the CHLAUTH record CHCKCLNT attribute. However, allowing unauthenticated remote access is not recommended.

CSQX791E

csect-name Client application *appl-name* from address *ip-address* did not supply a user ID and password, Detail: *detail*

Severity
8

Explanation

The client application *appl-name* running on host *ip-address* did not supply a user ID and password. The channel authentication (CHLAUTH) record for the connection requires a user ID and password, but none was supplied.

The active values of the channel were *detail*. The MATCH(RUNCHECK) mode of the DISPLAY CHLAUTH command can be used to identify the relevant CHLAUTH record.

System action

The channel is not started.

System programmer response

Ensure that the application provides a valid user ID and password, or change the queue manager connection authority (CONNAUTH) configuration to OPTIONAL to allow client applications to connect which have not supplied a user ID and password.

CSQX793E

csect-name The user ID and password for client application *appl-name* from address *ip-address* cannot be checked, Detail: *detail*

Severity
8

Explanation

The user ID and password for the client application *appl-name* running on host *ip-address* cannot be checked. The channel authentication (CHLAUTH) record for the connection requires an authentication check, but the queue manager is not configured to use connection authentication for clients.

The active values of the channel were *detail*. The MATCH(RUNCHECK) mode of the DISPLAY CHLAUTH command can be used to identify the relevant CHLAUTH record.

System action

The channel is not started.

System programmer response

Change the CHLAUTH configuration so that client authentication is not required, or alter the queue manager connection authority (CONNAUTH) configuration to enable client authentication checks.

CSQX797E

csect-name Unable to send message for channel *channel-name*, MQCC=*mqcc* MQRC=*mqrc (mqrc-text)*

Severity
8

Explanation

The send on channel *channel-name* could not be completed and the message could not be redirected to the dead-letter queue.

System action

The channel stops.

System programmer response

Refer to API completion and reason codes for information about *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form) to determine why the send failed.

Refer to previous messages to determine why the dead-letter queue is not available.

CSQX830I

csect-name Channel initiator active

Severity
0

Explanation

This is issued in response to the DISPLAY CHINIT command if the channel initiator is active.

CSQX831I

csect-name nn adapter subtasks started, *nn* requested

Severity
0

Explanation

This is issued in response to the DISPLAY CHINIT command, and shows how many adapter subtasks are currently active, and how many were requested by the CHIADAPS queue manager attribute. If the numbers differ, some adapter subtasks have failed and not been restarted, which could reduce processing capacity.

CSQX832I

csect-name nn dispatchers started, *nn* requested

Severity
0

Explanation

This is issued in response to the DISPLAY CHINIT command, and shows how many dispatchers are currently active, and how many were requested by the CHIDISPS queue manager attribute. If the numbers differ, some dispatchers have failed and not been restarted. The number of current TCP/IP and LU 6.2 channels allowed will be reduced proportionately, and other processing capacity may be reduced.

CSQX833I

csect-name nn SSL server subtasks started, *nn* requested

Severity
0

Explanation

This is issued in response to the DISPLAY CHINIT command, and shows how many SSL server subtasks are currently active, and how many were requested by the SSLTASKS queue manager attribute. If the numbers differ, some SSL server subtasks have failed and not been restarted, which could reduce processing capacity.

CSQX836I

csect-name nn Maximum channels - TCP/IP nn, LU 6.2 *nn*

Severity
0

Explanation

This is issued in response to the DISPLAY CHINIT command. It shows the maximum numbers of each type of channel that are allowed.

CSQX840I

csect-name nn channels current, maximum *nn*

Severity
0

Explanation

This is issued in response to the DISPLAY CHINIT command. It shows how many channels are current, and how many are allowed altogether, as requested by the MAXCHL queue manager attribute.

CSQX841I

csect-name nn channels active, maximum nn, including *nn* paused

Severity
0

Explanation

This is issued in response to the DISPLAY CHINIT command. Of the channels that are current, it shows how many are active (transmitting messages), and how many are allowed altogether to be active, by the ACTCHL queue manager attribute. It also shows how many of the active channels are paused, waiting to retry putting a message.

CSQX842I

csect-name nn channels starting, *nn* stopped, *nn* retrying

Severity
0

Explanation

This is issued in response to the DISPLAY CHINIT command. Of the channels that are current, it show how many are:

- waiting to become active, because the limit for active channels has been reached
- stopped, requiring manual intervention
- attempting to reconnect following a temporary error.

CSQX843I

csect-name TCP/IP listener INDISP=*disposition* retrying, for port *port* address *ip-address*

Severity
0

Explanation

This is issued in response to the DISPLAY CHINIT command for each TCP/IP listener that is trying to restart after an error. The channel initiator will attempt to restart the listener, at the intervals specified by the LSTRTMR queue manager attribute.

port and *ip-address* show the port and IP address combination on which it listens; if *ip-address* is '*', it listens on all available IP addresses. *disposition* shows which type of incoming requests the listener handles:

QMGR
those directed to the target queue manager

GROUP
those directed to the queue sharing group.

CSQX844I

csect-name LU 6.2 listener INDISP=*disposition* retrying, for LU name *name*

Severity
0

Explanation

This is issued in response to the DISPLAY CHINIT command for each LU 6.2 listener that is trying to restart after an error. The channel initiator will attempt to restart the listener at the intervals specified by the LSTRTMR queue manager attribute.

disposition shows which type of incoming requests the listener handles:

QMGR
those directed to the target queue manager

GROUP
those directed to the queue sharing group.

CSQX845I

csect-name TCP/IP system name is *name*

Severity
0

Explanation

This is issued in response to the DISPLAY CHINIT command, and shows the TCP/IP system name that is being used, as specified in the TCPNAME queue manager attribute.

CSQX846I

csect-name TCP/IP listener INDISP=*disposition* started, for port *port* address *ip-address*

Severity
0

Explanation

This is issued in response to the DISPLAY CHINIT command for each TCP/IP listener that is active.

port and *ip-address* show the port and IP address combination on which it listens; if *ip-address* is '*', it listens on all available IP addresses. *disposition* shows which type of incoming requests the listener handles:

QMGR
those directed to the target queue manager

GROUP
those directed to the queue sharing group.

CSQX847I

csect-name LU 6.2 listener INDISP=*disposition* started, for LU name *name*

Severity
0

Explanation

This is issued in response to the DISPLAY CHINIT command for each LU 6.2 listener that is active.

disposition shows which type of incoming requests the listener handles:

QMGR
those directed to the target queue manager

GROUP
those directed to the queue sharing group.

CSQX848I

csect-name TCP/IP listener INDISP=*disposition* not started

Severity
0

Explanation

This is issued in response to the DISPLAY CHINIT command for each TCP/IP listener that is not active.

disposition shows which type of incoming requests the listener handles:

QMGR
those directed to the target queue manager

GROUP
those directed to the queue sharing group.

System programmer response

If the listener had been started, and was not deliberately stopped, this might be because there was an error in the communications system. The channel initiator will attempt to restart the listener, at the intervals specified by the LSTRTMR queue manager attribute.

CSQX849I

csect-name LU 6.2 listener INDISP=*disposition* not started

Severity
0

Explanation

This is issued in response to the DISPLAY CHINIT command for each LU 6.2 listener that is not active.

disposition shows which type of incoming requests the listener handles:

QMGR
those directed to the target queue manager

GROUP
those directed to the queue sharing group.

System programmer response

If the listener had been started, and was not deliberately stopped, this might be because there was an error in the communications system. The channel initiator will attempt to restart the listener, at the intervals specified by the LSTRTMR queue manager attribute.

CSQX871I

csect-name Cluster maintenance has been running for *num-mins* minutes, phase *maintenance-phase* has so far processed *num-records* records

Severity
0

Explanation

A queue manager will periodically perform a maintenance cycle to refresh and remove state associated with the clusters it is a member of. This message gives an indication of the progress being made.

System action

For large clusters this maintenance process may take a significant period of time. In such situations this message will be periodically repeated until maintenance has completed, at which time message CSQX872I will be output.

CSQX872I

csect-name Cluster maintenance has completed after *num-mins* minutes, *num-records* records were processed

Severity
0

Explanation

A queue manager will periodically perform a maintenance cycle to refresh and remove state associated with the clusters it is a member of. This message follows one or more instances of message CSQX871I and indicates the cycle has completed.

System action

None

CSQX875I

csect-name REFRESH CLUSTER processing started for cluster cluster-name

Severity
0

Explanation

A REFRESH CLUSTER command has been issued on this queue manager.

In phase one this will discard all locally cached information for the cluster and request new information from other members of the cluster when necessary. Phase two processes the information received. For large cluster configurations this process can take a significant amount of time, especially on full repository queue managers. During this time applications attempting to access cluster resources may see failures to resolve cluster resources. In addition, cluster configuration changes made on this queue manager may not be processed until the refresh process has completed.

System action

Defer any cluster related work on this queue manager until both phases are complete.

Message CSQX442I or CSQX404I will be issued at the end of phase one.

Completion of phase two can be determined when the SYSTEM.CLUSTER.COMMAND.QUEUE has reached a consistently empty state.

CSQX876I

csect-name Cluster cache compression started

Severity
0

Explanation

Periodically cluster management will compress its local cache. Compression can take a significant period of time for certain operations, such as performing a CLUSTER REFRESH. During the compression task, cluster management commands will not be processed.

Once the compression task has completed message CSQX877I will be issued.

CSQX877I

csect-name Cluster cache compression completed

Severity
0

Explanation

The cluster cache compression activity, indicated by message CSQX876I, has now completed.

CSQX878I

csect-name Repository command error, command *command*, cluster object *object-name*, sender *sender-id*, reason *reason*

Severity
8

Explanation

An internal cluster repository command failed to complete successfully. Earlier messages in the log will contain details of the problem. Failure to successfully process a command can leave a cluster in an inconsistent state.

System action

Processing continues

System programmer response

If the problem cannot be resolved, collect the items listed in the Problem Determination section and contact your IBM® support center.

The *reason* is undocumented as it is intended for IBM Support only.

CSQX879E

csect-name Conflicting clustered topic *topic-name* from queue manager *qmgr-name*

Severity
8

Explanation

A conflict has been detected for clustered topic *topic-name*.

Two clustered topics conflict if any of the following conditions are true:

1. They have the same topic string but have a different topic name
2. They have the same topic string, or one is an ancestor of the other in the topic tree, and they have a different cluster name
3. They have the same topic string, or one is an ancestor of the other in the topic tree, and they have incompatible values for the cluster route attribute

System action

The CLSTATE attribute of the clustered topic identified by *topic-name* is set to INVALID and the topic is no longer used by the queue manager.

System programmer response

Review the clustered topics visible to the queue manager and correct any conflicts by modifying or deleting the definitions in error. After updating the topic definitions, ensure all clustered topics have a CLSTATE of ACTIVE on all queue managers in the same cluster.

CSQX888E

csect-name Cluster object *object-name* in cluster *cluster-name* (QMID *qmid*) expiring due to definition mismatch.

Severity
8

Explanation

Object *object-name* shared in cluster *cluster-name* (QMID *qmid*) has not been updated for at least 30 days but an alternative instance remains active in the cluster.

This often indicates an administrative error, such as recovery from backup of a cluster queue manager without REFRESH CLUSTER being issued.

If no action is taken the alternative definition will be accepted in place of the current definition before it expires.

System action

None

System programmer response

Review administrative procedures to ensure only a single instance of the queue manager with this QMID is active in the cluster at any given time.

If only the 'correct' host for this object is now active, you can optionally issue REFRESH CLUSTER on that queue manager to ensure its definitions are used from this point onwards and suppress further error messages.

CSQX889E

csect-name Cluster object *object-name* in cluster *cluster-name* (QMID *qmid*) replaced with 'older' definition.

Severity
8

Explanation

Object *object-name* shared in cluster *cluster-name* (QMID *qmid*) was due to expire, but publications from an alternative instance were received. To prevent expiry these definitions will be used, but this might indicate an operational error in the cluster has occurred.

System action

None

System programmer response

No action is required as the 'visible' definition will now be used, but you might want to review administrative procedures to ensure that only a single instance of the queue manager with this QMID is active in the cluster at any given time.

z/OS *Initialization procedure and general services messages (CSQY...)*

CSQY000I
IBM MQ for z/OS V*n release_type*

Explanation

This message is issued when the queue manager starts, and shows the release level and release type.

CSQY002I
QUEUE MANAGER STOPPING

Explanation

The STOP QMGR command is accepted. Message CSQ9022I is issued when the queue manager shutdown process has completed. The message is issued either to the originator of the STOP QMGR command, or to the z/OS console from which the START QMGR command was received.

System action

Queue manager shutdown is initiated.

CSQY003I
QUEUE MANAGER IS ALREADY ACTIVE

Explanation

The START QMGR command has not been accepted, because the queue manager is active. Message CSQ9023E is issued after this message.

CSQY004I
QUEUE MANAGER IS ALREADY STOPPING

Explanation

The STOP QMGR command has not been accepted either because the queue manager shutdown is in progress for the specified option (QUIESCE or FORCE), or because the QUIESCE option was specified after a FORCE option had been accepted previously. Message CSQ9023E is issued after this message.

System action

Queue manager shutdown continues.

CSQY005E
QUEUE MANAGER STARTUP TERMINATED, INVALID START COMMAND

Explanation

The queue manager can be started only by a START QMGR command.

System action

Queue manager startup is terminated.

CSQY006E
csect-name INVALID AMODE OR RMODE ATTRIBUTE FOUND FOR LOAD MODULE *module-name*

Explanation

The queue manager initialization procedures found that a module had an invalid AMODE or RMODE attribute when it was loaded. *module-name* is the name of the load module with an invalid addressing or residency mode.

System action

Queue manager startup terminates abnormally.

System programmer response

Verify that all installation and maintenance activities against IBM MQ have been done correctly. If you are unable to correct the problem, contact IBM Support.

CSQY007E
csect-name QUEUE MANAGER STARTUP TERMINATED, INVALID OPERATING SYSTEM LEVEL

Explanation

The queue manager initialization procedures found that the level of the operating system did not have the function required for correct queue manager operation.

System action

Queue manager startup terminates abnormally.

System programmer response

Verify that the prerequisite, or later, level of the operating system is installed. If you are unable to correct the problem, contact IBM Support.

CSQY008I
QUEUE MANAGER SHUTDOWN REQUEST NOT ACCEPTED

Explanation

The STOP QMGR command has not been accepted because startup has not completed to the point where shutdown can occur. Message CSQ9023E is issued after this message.

System action

Queue manager startup continues, and the STOP QMGR command is ignored.

CSQY009I
verb-name pkw-name COMMAND ACCEPTED FROM USER(*userid*), STOP MODE(*mode*)

Explanation

This message is issued to record who issued the command to stop IBM MQ, and what type of stop it was. *verb-name* might include the command prefix (CPF). This depends on how the command was entered.

CSQY010E
csect-name LOAD MODULE *module-name* IS NOT AT THE CORRECT RELEASE LEVEL

Explanation

The named load module is not at the correct level for the version of the queue manager that was being used.

System action

If detected by the queue manager, startup terminates abnormally with reason code X'00E80161'. If detected by the channel initiator (*module-name* is CSQXJST), it does not start.

If detected by the AMS enablement module (DRQ0NABL), the queue manager only fails to start if SPLCAP=YES is specified in the system parameters. In this case message CSQY029E is issued.

System programmer response

Verify that the correct IBM MQ program libraries are being used (for the queue manager or channel initiator as appropriate) and that all installation and maintenance activities againstIBM MQ have been done correctly. If the early processing program is incorrect (*module-name* is CSQ3EPX), refresh it by issuing the REFRESH QMGR TYPE(EARLY) command.

If you are unable to correct the problem, contact IBM Support.

CSQY011E
csect-name COMMAND PREFIX REGISTRATION FAILED. INVALID CHARACTER(S) IN CPF

Explanation

Command prefix registration failed because the command prefix (CPF) contains invalid characters.

System action

The queue manager does not start.

System programmer response

Reissue the z/OS command SETSSI ADD with the correct CPF parameter. Correct the CPF parameter in the record of SYS1.PARMLIB member IEFSSNxx. For information about the parameters, see Updating the subsystem name table.

CSQY012E
csect-name COMMAND PREFIX REGISTRATION FAILED. INVALID CHARACTER(S) IN QUEUE MANAGER NAME

Explanation

Command prefix registration failed because the queue manager name used as the owner of the command prefix (CPF) contains invalid characters.

System action

The queue manager does not start.

System programmer response

Reissue the z/OS command SETSSI ADD with the correct CPF parameter. Correct the CPF parameter in the record of SYS1.PARMLIB member IEFSSNxx. For information about the parameters, see Updating the subsystem name table.

CSQY013E
csect-name COMMAND PREFIX REGISTRATION FAILED. CPF ALREADY DEFINED

Explanation

Command prefix registration failed because the command prefix (CPF) was already defined to z/OS.

System action

The queue manager does not start.

System programmer response

Reissue the z/OS command SETSSI ADD with the correct CPF parameter. Correct the CPF parameter in the record of SYS1.PARMLIB member IEFSSNxx. For information about the parameters, see Updating the subsystem name table.

CSQY014E
csect-name COMMAND PREFIX REGISTRATION FAILED. CPF IS A SUBSET OF A CPF ALREADY DEFINED

Explanation

Command prefix registration failed because the command prefix (CPF) is a subset of a CPF already defined to z/OS.

System action

The queue manager does not start.

System programmer response

Reissue the z/OS command SETSSI ADD with the correct CPF parameter. Correct the CPF parameter in the record of SYS1.PARMLIB member IEFSSNxx. For information about the parameters, see Updating the subsystem name table.

CSQY015E
csect-name COMMAND PREFIX REGISTRATION FAILED. CPF IS A SUPERSET OF A CPF ALREADY DEFINED

Explanation

Command prefix registration failed because the command prefix (CPF) is a superset of a CPF already defined to z/OS.

System action

The queue manager does not start.

System programmer response

Reissue the z/OS command SETSSI ADD with the correct CPF parameter. Correct the CPF parameter in the record of SYS1.PARMLIB member IEFSSNxx. For information about the parameters, see Updating the subsystem name table.

CSQY016E
csect-name SYSTEM ERROR DURING COMMAND PREFIX REGISTRATION

Explanation

A z/OS error occurred during command prefix (CPF) registration.

System action

The queue manager does not start.

System programmer response

Check the z/OS console for other messages relating to the problem.

CSQY017E
csect-name INCORRECT STORAGE PROTECT KEY

Explanation

The queue manager initialization procedures found that the storage protect key was not 7. The most likely causes are that the program properties table (PPT) entry for CSQYASCP has not been specified correctly, or that the IBM MQ program libraries or other libraries in the IBM MQ STEPLIB are not APF authorized.

System action

Queue manager startup terminates abnormally with reason code X'00E80162'.

System programmer response

Check that all the libraries you include in the IBM MQ STEPLIB have been APF-authorized. Also, ensure that you use the actual library name and not the data set alias of the libraries in APF list.

For information about specifying the PPT entry for CSQYASCP and about APF authorization for the IBM MQ program libraries, see Updating the z/OS program properties table.

CSQY018E
csect-name INCORRECT APF AUTHORIZATION

Explanation

The queue manager initialization procedures found that they were not APF authorized. The most likely cause is that one or more of the data sets in the //STEPLIB concatenation is not APF authorized.

System action

Queue manager startup terminates abnormally with reason code X'00E80163'.

System programmer response

Check all the libraries that you include in the IBM MQ STEPLIB are APF-authorized. Also, check that you do not use a data set alias of the libraries in the APF list, use the actual library name instead.

For information about APF authorization for the IBM MQ program libraries, see APF authorize the IBM MQ load libraries.

CSQY019E
csect-name QUEUE MANAGER STARTUP TERMINATED, INVALID PARAMETER MODULE LEVEL, REBUILD *macro-name*

Explanation

The queue manager initialization procedures found that the level of the parameter module (named in the preceding CSQY001I message) is not at the correct level for this version of the queue manager.

System action

Queue manager startup terminates abnormally with reason code 00E80051.

System programmer response

Rebuild the parameter module ensuring that *macro-name* is recompiled with the same level of code that the queue manager is running with.

For more information about the macros used to build the parameter module see, Task 17: Tailor your system parameter module.

CSQY020E
csect-name CHANNEL INITIATOR STARTUP TERMINATED, INVALID START COMMAND

Explanation

The channel initiator can be started only by a **START CHINIT** command.

System action

Channel initiator startup is terminated.

System programmer response

Start the channel initiator using the **START CHINIT** command

CSQY021E
csect-name QUEUE MANAGER STARTUP TERMINATED, INSUFFICIENT MEMLIMIT

Explanation

The queue manager initialization procedures found that the configured MEMLIMIT is less than 512MB.

System action

Queue manager startup terminates abnormally.

CSQY022I
QUEUE MANAGER INITIALIZATION COMPLETE

Explanation

This message is issued when the initialization of the queue manager completes normally, and it is ready for use.

CSQY023A
SOME OBJECTS COULD NOT BE MIGRATED, MANUAL RESOLUTION REQUIRED. REPLY TO ACKNOWLEDGE AND CONTINUE STARTUP

Explanation

The queue manager has detected that it was previously running at an earlier version and forward migration has been performed. However, some objects could not be migrated because of locks held by in-doubt transactions. Message CSQI970E is also issued for each object that could not be migrated.

This message is not issued during subsequent restarts of the queue manager whilst it is running at the same version.

System action

Startup is suspended and the queue manager waits for the operator to reply with any single character.

System programmer response

Reply to acknowledge this message and allow queue manager startup to proceed.

Thereafter, additional action is required to complete forward migration of each identified object.

For more information see the description of message CSQI970E.

CSQY024I
AMS not started, product usage is not set.

Severity
8

Explanation

The system parameter SPLCAP is set to YES in the queue manager's ZPARM, however, AMSPROD has not been set or QMGRPROD not set to ADVANCEDVUE.

System action

Queue manager startup is terminated.

System programmer response

If Advanced Message Security is required, set the appropriate value for AMSPROD or QMGRPROD, based on your product entitlement. See product usage recording with IBM MQ for z/OS products.

Alternatively, if IBM MQ Advanced Message Security is not required, update the queue manager's ZPARM, to set SPLCAP to NO.

CSQY025I
AMS is enabled.

Severity
0

Explanation

This message indicates that IBM MQ Advanced Message Security is enabled.

System action

Queue manager startup continues.

System programmer response

None.

CSQY027I
csect-name AMS STARTING

Severity
0

Explanation

The Advanced Message Security (AMS) address space has been started because the system parameter SPLCAP is set to YES in the queue manager's ZPARM.

System action

Connections to the queue manager are permitted, but MQI calls that might require AMS function are suspended until AMS is available. Further messages are output when the AMS feature initializes.

CSQY028I
csect-name AMS HAS STARTED

Severity
0

Explanation

Advanced Message Security (AMS) initialization has completed successfully.

System action

Applications waiting for AMS function are resumed.

CSQY029E
csect-name QUEUE MANAGER STARTUP TERMINATED, AMS INITIALIZATION FAILED

Severity
12

Explanation

A severe error occurred during initialization of Advanced Message Security (AMS).

System action

The queue manager abnormally terminates with abend code 6C6 and reason 00F00003.

System programmer response

Investigate the problem reported by preceding messages in the job log for the AMS address space (xxxxAMSM). Resolve the problem, then restart the queue manager. If you are unable to resolve the error, contact IBM Support.

CSQY030E
csect-name QUEUE MANAGER TERMINATING, AMS NOT AVAILABLE

Severity
12

Explanation

The Advanced Message Security (AMS) address space has ended abnormally due to an unrecoverable error.

System action

The queue manager abnormally terminates with abend code 6C6 and reason 00F00003.

System programmer response

Investigate the problem reported by preceding messages in the job log for the AMS address space (xxxxAMSM). Resolve the problem, then restart the queue manager. If you are unable to resolve the error, contact IBM Support.

CSQY031I
csect-name QUEUE MANAGER WAITING FOR AMS INITIALIZATION

Severity
0

Explanation

The Advanced Message Security (AMS) address space has been started because the system parameter SPLCAP is set to YES in the queue manager's ZPARM. This message is periodically issued until AMS initialization completes.

System action

Processing continues. Connections to the queue manager are permitted, but MQI requests that might require AMS function are suspended until AMS is available.

System programmer response

Investigate the delay in initializing Advanced Message Security by reviewing the messages output in the job log for the AMS address space (xxxxAMSM).

CSQY032E
csect-name QUEUE MANAGER STARTUP TERMINATED, UNABLE TO START AMS

Severity
12

Explanation

The queue manager attempted to start the Advanced Message Security (AMS) address space because the system parameter SPLCAP is set to YES in the queue manager's ZPARM. The AMS address space (xxxxAMSM) failed to start, which might be because another job with the same name is active, or there is an error in the started task JCL.

System action

The queue manager abnormally terminates with abend code 6C6 and reason 00F00003.

System programmer response

Investigate why the AMS address space could not be started. Resolve the problem by terminating an existing address space if one is active, or correct the started task JCL if required, then restart the queue manager.

CSQY033A
csect-name QUEUE MANAGER NOT AVAILABLE, AMS INITIALIZATION ERROR

Severity
12

Explanation

A severe error occurred during initialization of Advanced Message Security (AMS).

System action

Queue manager startup is interrupted. The queue manager accepts commands, but MQI requests that might require AMS function fail with reason code 2063 (MQRC_SECURITY_ERROR).

System programmer response

Investigate the problem reported by preceding messages in the job log for the AMS address space (xxxxAMSM). Resolve the problem, then shutdown and restart the queue manager. If you are unable to resolve the error, contact IBM Support.

CSQY034I
csect-name QUEUE MANAGER WAITING FOR AMS TO SHUTDOWN

Severity
0

Explanation

The queue manager is stopping and has requested the Advanced Message Security (AMS) address space (xxxxAMSM) ends. This message is periodically issued until AMS shutdown completes.

System action

The queue manager continues to wait for the AMS address space to end.

System programmer response

If this message is repeatedly issued examine the job log for the AMS address space to determine why it has not ended. If the problem cannot be resolved terminate the address space to allow queue manager shutdown to continue.

CSQY035I
csect-name AMS HAS SHUTDOWN

Severity
0

Explanation

The Advanced Message Security (AMS) address space (xxxxAMSM) has ended.

System action

Queue manager shutdown continues.

CSQY036I
QMGRPROD= *prod-value*, recording product usage for *product-name*, product ID *product-id*

Explanation

This message is issued when the queue manager starts if SMF 89 product usage records are to be recorded by the queue manager. *product-name* is the descriptive name of the product, and *product-id* is the product ID that is to be used in the SMF 89 data. For example:

- QMGRPROD=MQ, recording product usage for IBM MQ for z/OS, product ID 5656-MQA - this indicates that usage data is to be collected for the IBM MQ for z/OS product.
- QMGRPROD=ADVANCEDVUE, recording product usage for IBM MQ Advanced for z/OS Value Unit Edition, product ID 5656-AVA - this indicates that usage data is to be collected for the IBM MQ Advanced for z/OS Value Unit Edition product.

See *z/OS MVS Product Management* for more information on product usage recording.

CSQY037I
Product usage data is not being recorded for *product-name*, product ID *product-id*

Explanation

This message is issued when the queue manager starts, if SMF 89 product usage data is not being recorded by the queue manager. This might be because SMF 89 collection is not activated for the system.

See *z/OS MVS Product Management* for more information on product usage recording.

CSQY038E
csect-name QUEUE MANAGER STARTUP TERMINATED, *product* is not valid for *prod-keyword* in *prod-source*

Explanation

The queue manager initialization procedures found a value *product* for *prod-keyword* in *prod-source* that is not valid.

prod-keyword can be QMGRPROD or AMSPROD, and *prod-source* can be START COMMAND, JCL PARM or CSQ6USGP.

The message can be issued more than once, if more than one value, that is not valid, is found.

System action

Queue manager startup terminates abnormally with reason code 00E80010.

System programmer response

Correct the value that is not valid:

- If prod-source is START COMMAND, see START QMGR for further information.
- If prod-source is JCL PARM, see Using MQSC to start and stop a queue manager on z/OS for information on coding the JCL parameter for the queue manager JCL.
- If prod-source is CSQ6USGP, see Using CSQ6USGP for information on configuring values using CSQ6USGP.

CSQY039I

Backwards migration is supported to Version `v.r.m`

Explanation

The queue manager has previously been started up using the indicated earlier version of IBM MQ. If required it is possible to start the queue manager up again using that earlier version if the START QMGR BACKMIG(VRM) command has first been run against the current version.

CSQY040I

Backwards migration not supported

Explanation

The queue manager cannot be started using an earlier version of IBM MQ. This message is output if the queue manager:

- Has been created using the current version.
- Was previously started, using a Continuous Delivery version with a non-zero modification number, for example, IBM MQ 9.2.4, before being migrated to the current version.
- Was previously started using a version of IBM MQ to which the current version does not support backwards migration.

CSQY041D

Starting queue manager at a CD release will prevent backward migration. Reply Y to continue, N to cancel

Explanation

This message is issued as a `write to operator with reply` (WTOR) when the queue manager detects that it is being migrated from an LTS release, such as IBM MQ 9.2.0, or a CD release, with a modifier value of 0 such as IBM MQ 9.3.0 CD, to a CD release with a non-zero modifier value, such as IBM MQ 9.3.1

CD releases with non zero modifiers do not support backwards migration, so this message is issued to confirm that the IBM MQ administrator is aware of this, and does want to migrate from LTS to CD.

Attention: Once the queue manager starts at CD it will not be able to migrate back to the earlier release.

The queue manager will not complete start up until you reply to the WTOR.

To confirm that the queue manager should migrate to CD, reply to the WTOR with the letter Y. The queue manager then starts up as normal and issues message CSQY040I.

To cancel migration, reply to the WTOR with the letter N. The queue manager will abend with abend code 5C6 and reason code 00E80171, and shutdown. You can then start up the queue manager using the libraries that it was previously using.

If you reply to the WTOR with anything other than the letter N or Y, the message is reissued until a correct reply is received.

System action

The queue manager waits until a valid response to the WTOR is provided, at which point it will either continue startup or terminate.

System programmer response

Reply to the WTOR with either the letter Y or N.

CSQY042E
Backward migration to v.r.m not supported

Explanation

The queue manager was started with parameter BACKMIG(vrm), but backward migration to the earlier version is not supported.

System action

Queue manager start up terminates abnormally with reason code X'00E80084'.

System programmer response
Check for message CSQY039I or message CSQY040I to determine if backward migration is permitted.

CSQY043E
Backward migration to v.r.m failed

Explanation

The queue manager was started with parameter BACKMIG(vrm).

Backward migration to the earlier version was attempted, but failed unexpectedly.

System action

Queue manager start up terminates abnormally with a reason code indicating the problem.

System programmer response

Investigate and correct the cause of failure and retry the command.

CSQY044D
Page set or sets offline during backward migration. Reply Y to continue backward migration or N to cancel.

Explanation

The queue manager was started with parameter BACKMIG(vrm).

Backward migration to the earlier version was attempted, but at least one page set that has been used is currently offline. This message is proceeded by an instance of message CSQI005I for each offline page set.

The queue manager will not be able to access any corresponding page set after it has been backward migrated without manual intervention.

System action

The queue manager waits until a valid response to the WTOR is provided, at which point it will either continue backward migration or end.

System programmer response

Reply N to cancel backward migration and end the queue manager. Resolve the offline page set or page sets and retry the command.

Reply Y to continue backward migration without the page set or sets. You might be able to migrate the page set or sets manually, or recover each to a new data set later using the CSQUTIL FORMAT command.

CSQY045I
Backward migration to v.r.m completed

Explanation

The queue manager was started with parameter BACKMIG(vrm), and has completed backward migration to the earlier release.

System action

The queue manager is shutdown.

System programmer response

Restart the queue manager at the earlier release.

CSQY046I

Channel *channel-name* has a CONNAME >48 chars, this will be blanked

Explanation

A channel with a **CONNAME** longer than 48 characters has been detected during backwards migration to an IBM MQ version that only supports a maximum of 48 characters.

If the system programmer decides to proceed with backwards migration, see message CSQY047D, then the **CONNAME** in the channel will be blanked out. The channel will not be useable until it is altered to use a valid **CONNAME**.

System action

Continue processing all channel definitions. When processing is completed, issue message CSQY047D so the system programmer can decide whether to continue, or abort, backwards migration.

System programmer response

Wait for message CSQY047D to be issued and respond to it.

CSQY047D

num channels have a CONNAME >48 chars, these will be blanked. Reply Y to continue or N to cancel

Explanation

This message is issued as a write to operator with reply (WTOR). One or more CSQY046I messages have been issued. If the system programmer wants to abort backwards migration, they can reply to the WTOR with the letter N. The system programmer can restart the queue manager at the current version and adjust the channels to use a **CONNAME** of up to 48 characters, or delete the channels, and reattempt backwards migration.

If the system programmer is happy for backwards migration to continue, they can reply with the letter Y. Once backwards migration has completed the channels cannot be used until the **CONNAME**s are set to a valid value that is up to 48 characters.

If the system programmer replies to the WTOR with anything other than the letter N or Y, the message is reissued until a correct reply is received.

System action

If the reply is N, stop backwards migration and abend. The queue manager can then be started up at the current version.

If the reply is Y, continue backwards migration and adjust the **CONNAME**s of the affected channels.

System programmer response

Respond to the WTOR with either Y or N and then perform the remedial actions in the Explanation section.

CSQY048I

CHLAUTH record *record-name* contains an ADDRESS >48 chars, this will be removed

Explanation

A channel authentication record with an **ADDRESS** longer than 48 characters has been detected during backwards migration to an IBM MQ version that only supports a maximum of 48 characters.

If the system programmer decides to proceed with backwards migration, see message CSQY049D, then the record will be removed. After backwards migration a new record should be created.

System action

Continue processing all channel authentication records. When processing is completed, issue message CSQY049D so the system programmer can decide whether to continue, or abort, backwards migration.

System programmer response

Wait for message CSQY049D to be issued and respond to it.

CSQY049D

num CHLAUTH records have an ADDRESS >48 chars, these will be removed. Reply Y to continue or N to cancel

Explanation

This message is issued as a write to operator with reply (WTOR). One or more CSQY048I messages have been issued. If the system programmer wants to abort backwards migration, they can reply to the WTOR with the letter N. The system programmer can restart the queue manager at the current version and then adjust the channel authentication records to use an **ADDRESS** of up to 48 characters, or remove the records, and reattempt backwards migration.

If the system programmer is happy for backwards migration to continue, they can reply with the letter Y. Once backwards migration has completed a new record should be created.

If the system programmer replies to the WTOR with anything other than the letter N or Y, the message is reissued until a correct reply is received.

System action

If the reply is N, stop backwards migration and abend. The queue manager can then be started up at the current version.

If the reply is Y, continue backwards migration and remove the affected channel authentication records.

System programmer response

Respond to the WTOR with either Y or N and then perform the remedial actions in the Explanation section.

CSQY100I

csect-name SYSTEM parameters ...

Explanation

The queue manager is being started with the system parameter values shown in the following messages.

System action

Queue manager startup processing continues.

CSQY101I

CSQY102I, CSQY103I, CSQY104I, CSQY105I, CSQY106I, CSQY107I, CSQY108I, CSQY109I, CSQY130I: *csect-name* parms

Explanation

This series of messages shows the system parameter values that the queue manager is using. (Some values are followed by their internal hexadecimal representation in parentheses.) For information about the system parameters for the CSQ6SYSP macro, see Using CSQ6SYSP.

System action

Queue manager startup processing continues.

CSQY110I

csect-name LOG parameters ...

Explanation

The queue manager is being started with the log parameter values shown in the following messages.

System action

Queue manager startup processing continues.

CSQY111I

CSQY112I, CSQY113I, CSQY114I: *csect-name* parms

Explanation

This series of messages shows the log parameter values that the queue manager is using. For information about the log parameters in the CSQ6LOGP macro, see Using CSQ6LOGP.

System action

Queue manager startup processing continues.

CSQY120I
csect-name ARCHIVE parameters ...

Explanation

The queue manager is being started with the archive parameter values shown in the following messages.

System action

Queue manager startup processing continues.

CSQY121I
CSQY122I, CSQY123I, CSQY124I: *csect-name* parms

Explanation

This series of messages shows the archive parameter values that the queue manager is using. For information about the archive parameters in the CSQ6ARVP macro, see Using CSQ6ARVP.

System action

Queue manager startup processing continues.

CSQY140I
csect-name USAGE parameters

Explanation

The queue manager is being started with the usage parameter values shown in the following messages.

These values can be overridden by values provided in the queue manager JCL or on the START QMGR command. The resolved values are shown in message CSQY037I and CSQ0619I.

CSQY141I
csect-name No USAGE parameters provided

Explanation

No queue manager usage parameters are provided, and the defaults are assumed.

These values can be overridden by values provided in the queue manager JCL or on the START QMGR command. The resolved values are shown in message CSQY037I and CSQ0619I.

CSQY142I
csect-name **parms**

Explanation

This message shows the usage parameter values that the queue manager is using. For information about the usage parameters for the CSQ6USGP macro, see Using CSQ6USGP.

CSQY200E
csect-name ARM *request-type* for element *arm-element* type *arm-element-type* failed, rc=*rc* reason=*reason*

Explanation

An ARM request (IXCARM REQUEST=*request-type*) for the specified element failed. *rc* is the return code and *reason* is the reason code (both in hexadecimal) from the call.

System action

None.

System programmer response

See the *z/OS MVS Programming Sysplex Services Reference* manual for information about the Return and reason codes from the IXCARM call.

If you are unable to solve the problem, contact IBM Support.

CSQY201I
csect-name ARM REGISTER for element *arm-element* type *arm-element-type* successful

Explanation

The specified element was successfully registered with ARM.

System action

None.

CSQY202E
csect-name ARM registration failed

Explanation

An attempt to register with ARM failed.

System action

Processing continues, but automatic restart is not available.

System programmer response

See the preceding CSQY200E message for more information about the failure.

CSQY203E
csect-name ARM *request-type* for element *arm-element* type *arm-element-type* timed out, rc=*rc* reason=*reason*

Explanation

An ARM request (IXCARM REQUEST=*request-type*) was issued but some predecessor element specified in the ARM policy did not issue an ARM READY request within its specified time interval.

System action

Processing continues.

System programmer response

None required. However, if your program cannot run without the predecessor element, some installation-defined action might be necessary.

CSQY204I
csect-name ARM DEREGISTER for element *arm-element* type *arm-element-type* successful

Explanation

The specified element was successfully deregistered from ARM.

System action

None.

CSQY205I
csect-name ARM element *arm-element* is not registered

Explanation

A STOP QMGR command requested ARM restart, but the queue manager was not registered for ARM.

System action

The queue manager stops normally, but will not be automatically restarted.

System programmer response

Restart the queue manager manually.

CSQY210E
csect-name call-name call for name name-token failed, rc=*rc*

Explanation

During processing for a group connect, a name token services call failed. *rc* is the return code (in hexadecimal) from the call.

System action

If the failure occurs in the batch adapter (*csect-name* CSQBCON or CSQBDSC), the application call will fail with a reason code of MQRC_UNEXPECTED_ERROR. Otherwise (*csect-name* CSQYGRA1), processing continues, but the group connect facility will not be available.

System programmer response

Go to the appropriate volume of the *z/OS MVS Programming: Assembler Services Reference* manual for information about the return codes:

- IEANTRT
- IEANTCR
- IEANTDL

from the name token services call.

If you are unable to solve the problem, take a stand-alone system dump and contact IBM Support.

CSQY211I
csect-name Unable to add entry to group connect name table (at *table-addr*)

Explanation

During initialization for the group connect facility, a new entry could not be added to the name table for this queue manager. The most likely cause is that there is already the maximum of 32 queue managers active in the group.

System action

Processing continues, but this queue manager will not be available for group connection.

System programmer response

Reduce the number of active queue managers and restart this queue manager. If this does not solve the problem, contact IBM Support.

CSQY212E
csect-name Unable to find the group attach table

Explanation

During initialization for the group connect facility, the group attach table could not be found. The most likely causes are that an error occurred during subsystem initialization, or that the subsystem was not initialized with the latest version of the IBM MQ early code.

System action

Processing continues, but the group connect facility will not be available to CICS.

System programmer response

Ensure that the libraries with the latest version, release, or maintenance level of the IBM MQ early code are in the libraries used for the z/OS LPA, and refresh the early code for the queue manager using

the IBM MQ command REFRESH QMGR TYPE(EARLY). See the Task 3: Update the z/OS link list and LPA.

CSQY220I
csect-name Queue manager storage usage : local storage : used *mm*MB, free *nn*MB : above bar : used *aabb*,free *cc*

Explanation

This message displays the amount of virtual storage currently used and available:

- in the extended private region (local storage).
- above the Bar (64 bit storage).

  The amount of storage used is displayed in the most appropriate unit (MB / GB) according to the number of bytes, and are approximations. If the amount of storage available exceeds 10 GB, ' >10 GB' is displayed. In all other cases the amount of storage available is displayed in the most appropriate unit. For the amount of storage space available, the total is rounded down to a whole number in the appropriate unit (MB /GB). For example, if the value of 3 GB is displayed the amount of free storage is greater than or equal to 3 GB and less than 4 GB.

This message is logged at queue manager start and then either every hour if the usage does not change or when the memory usage changes (up or down) by more than 2%.

The message is also generated if the ALTER BUFFPOOL command makes a change to the value either for LOCATION, or BUFFERS.

System action

Processing continues. Any special actions taken by IBM MQ or that are required, are indicated by the CSQY221I and CSQY222E messages.

System programmer response

No action is required at this time. However, a frequent occurrence of this message might be an indication that the system is operating beyond the optimum region for the current configuration.

CSQY221I
csect-name Queue manager is short of local storage

Explanation

The queue manager is running short of virtual storage in the extended private region.

System action

Processing continues. Storage contraction processing is performed, which attempts to remove unused storage from internal subpools so that it can be reused in other subpools. This might be necessary after a temporary need for a large amount of storage; for example, an unusually large unit of work being performed.

System programmer response

If only a few of these messages are output then no action is required at this time. However, a frequent occurrence of this message may be an indication that the system is operating beyond the optimum region for the current configuration and should be investigated.

CSQY222E
csect-name Queue manager is critically short of local storage - take action

Explanation

The queue manager is running critically short of virtual storage in the extended private region. Action should be taken to relieve the situation, and to avoid the possible abnormal termination of the queue manager.

System action

Processing continues. Storage contraction processing has been performed, but the remaining unallocated virtual storage is less than a predetermined safe amount. If storage use continues to increase, the queue manager might terminate abnormally in an unpredictable way.

System programmer response

Virtual storage is over-allocated for the current configuration. The following actions can reduce the virtual storage requirement:

- For buffer pools that have the LOCATION parameter set to BELOW, you can reduce buffer pool sizes with the ALTER BUFFPOOL command. Buffer pool statistics can be used to determine buffer pools which are over-allocated.
- Reduce the number of concurrent connections to the queue manager. The DISPLAY CONN command can be used to determine connections which are consuming queue manager resources.

If the problem persists after taking actions described above, it might be an indication of an internal error where storage is not freed (a 'storage leak'). If you suspect this, then collect at least two system dumps of the queue manager, separated by an interval of time, and contact your IBM support center.

CSQY223I
csect-name Queue manager is no longer short of local storage

Explanation

The queue manager is no longer short of virtual storage in the extended private region.

System action

Processing continues. Storage contraction processing has been performed, and the remaining unallocated virtual storage is more than a predetermined safe amount.

CSQY224I
csect-name Queue manager is short of local storage above the bar

Explanation

The queue manager is running short of virtual storage above the bar.

System action

Processing continues. Storage contraction processing is performed, which attempts to remove unused storage from internal subpools so that it can be reused in other subpools. This might be necessary after a temporary need for lots of storage; for example, more than the usual number of messages held on an indexed queue, or an unusually large unit of work being performed.

CSQY225E
csect-name Queue manager is critically short of local storage above the bar - take action

Explanation

The queue manager is running critically short of virtual storage above the bar. Action should be taken to relieve the situation, and to avoid the possible abnormal termination of the queue manager.

System action

Processing continues. Storage contraction processing has been performed, but the remaining unallocated virtual storage is less than a predetermined safe amount. If storage use continues to increase, the queue manager might terminate abnormally in an unpredictable way.

CSQY226I
csect-name Queue manager is no longer short of local storage above the bar

Explanation

The queue manager is no longer short of virtual storage above the bar.

System action

Processing continues. Storage contraction processing has been performed, and the remaining unallocated virtual storage is more than a predetermined safe amount.

CSQY227E
csect-name Unable to allocate storage above the bar using IARV64, RC=rc, reason=reason

Explanation

A request by the queue manager to allocate storage above the bar failed. rc is the return code and reason is the reason code (both in hexadecimal) from the z/OS IARV64 service.

System action

The queue manager will attempt to recover from the error. If recovery is not possible an application or queue manager abend, for example 5C6-00A30042, 5C6-00A31000 or 5C6-00E20045, will occur.

CSQY228E
ACE pool cannot be extended, ACELIM reached

Explanation

The internal storage pool used to manage control blocks representing new connections to the queue manager has reached the limit defined by the ACELIM system parameter.

System action

Queue manager processing continues. New connection requests might have failed, message CSQ3202E or CSQM078E give further information about the affected jobs.

System programmer response

Review the configured ACELIM value. It might be useful to use a STATISTICS CLASS(2) trace to establish the normal size of the ACE pool.

See Address space storage for more information.

CSQY270E

csect-name UNRECOGNIZED MESSAGE NUMBER *message-id*

Severity
8

Explanation

An unsuccessful attempt has been made to issue the message *message-id*. This message is issued only if the requested message could not be found in the IBM MQ message directory.

System action

Processing continues as though the requested message had been issued.

System programmer response

Use the message number (*message-id*) and look up the message in this product documentation. If you are using a language other than US English, ensure that you have installed the language feature correctly and that you have the appropriate load library data set concatenations in your job. Apart from that possibility, this might be an MQ system problem; see Troubleshooting and support.

Note: Messages are also used to provide text for constructing panels and reports. If such a message cannot be found, message CSQY270E will appear on the panel or report, generally in truncated form.

CSQY271I

MESSAGE GENERATOR INITIALIZATION PARAMETERS NOT FOUND. DEFAULTS ASSUMED

Severity
4

Explanation

The message generator was unable to access the routing code initialization parameter defined by the CSQ6SYSP macro. Default values defined by that macro are assumed.

System action

Queue manager initialization continues.

System programmer response

It might be necessary to change the CSQ6SYSP macro. For information about the system parameters for the CSQ6SYSP macro, see Using CSQ6SYSP.

CSQY290E

csect-name NO STORAGE AVAILABLE

Severity
4

Explanation

There was insufficient storage available for a system routine. *csect-name* shows the system routine function:

CSQAXDPS, CSQVXDPS
User exits (other than channel)

CSQXARMY
Channel initiator automatic restart

CSQXDCTS, CSQXTRPG
Channel initiator trace

CSQXDMPS
Channel initiator system dump

CSQXLDXS
User channel exits

CSQ2GFRR, CSQ2MFRR
IMS bridge system dump

System action

Processing continues, but the function provided by the system routine will be inhibited. For example, if the routine is CSQXLDXS, then user channel exits will not be available, and channels that use them will not start.

System programmer response

If the problem occurs in the queue manager, increase the size of the its address space, or reduce the number of queues, messages, and threads being used.

If the problem occurs in the channel initiator, increase the size of the its address space, or reduce the number of dispatchers, adapter subtasks, SSL server subtasks, and active channels being used.

CSQY291E

csect-name SDUMPX FAILED, RC=0000*ssrr*, *dump-identifier*

Severity
4

Explanation

The system dump routine was unable to issue a dump; the dump identifier was as shown in the message. *rr* is the return code and *ss* is the reason code (both in hexadecimal) from the z/OS SDUMPX service.

Usually the return code is 08. The most common reason codes for return code 08 are:

02
An SVC dump was suppressed because the capture phase of another SVC dump was in progress.

04
An SVC dump was suppressed by a SLIP NODUMP command.

0B
An SVC dump was suppressed by DAE.

System action

Processing continues.

System programmer response

Select the appropriate volume of the *z/OS MVS Authorized Assembler Services Reference* manual for information about the return code and reason code from the SDUMPX request.

For reason code 0B, that is, in the case of DAE suppression, see generating a suppressed dump.

To summarize, use IPCS option 3.5 , that is, `Utilities -> DAE` to set the T (TAKEDUMP) option for the dump symptom, or symptoms, you want to allow.

CSQY330I
Queue manager has restricted functionality

Explanation

The installation and customization options chosen for IBM MQ do not allow all functions to be used.

System action

Queue manager startup processing continues.

CSQY331E
parm value not allowed - restricted functionality

Explanation

The value specified for the *parm* system parameter is not allowed because the installation and customization options chosen for IBM MQ do not allow all functions to be used.

System action

The queue manager does not start.

CSQY332I
IMS Bridge not available - restricted functionality

Explanation

The IBM MQ-IMS bridge cannot operate because the installation and customization options chosen for IBM MQ do not allow all functions to be used.

System action

The MQ-IMS bridge does not start.

CSQY333E
Command not allowed - restricted functionality

Explanation

The command that was issued is not allowed because the installation and customization options chosen for IBM MQ do not allow all functions to be used.

System action

The command is ignored.

CSQY334E
csect-name keyword(*value*) not allowed - restricted functionality

Explanation

The value specified for the keyword is not allowed because the installation and customization options chosen for IBM MQ do not allow all functions to be used.

System action

The command is ignored.

System programmer response

CSQY335E
csect-name Channel *channel-name* unusable - restricted functionality

Explanation

The channel cannot be used because the installation and customization options chosen for IBM MQ do not allow all functions to be used.

System action

The requested operation fails.

CSQY336E
csect-name keyword not allowed - restricted functionality

Explanation

The keyword is not allowed because the installation and customization options chosen for IBM MQ do not allow all functions to be used.

System action

The command is ignored.

CSQY337E
csect-name keyword value length not allowed - restricted functionality

Explanation

The length of the value specified for the keyword is not allowed because the installation and customization options chosen for IBM MQ do not allow all functions to be used.

System action

The command is ignored.

CSQY340E
Queue manager has restricted functionality, but previously had full functionality. Unsupported objects will be deleted (losing messages), invalid attributes will be changed

Explanation

The installation and customization options chosen for IBM MQ do not allow all functions to be used. However, the queue manager has run previously without any functional restriction, and so might have objects and attribute settings that are not allowed with the restricted functionality.

In order to continue, these objects must be deleted (which might mean that messages are lost) and the attributes must be changed. The queue manager does this automatically.

System action

Message CSQY341D is issued and the operator's reply is awaited.

System programmer response

The operator has two options:

- Allow the queue manager to delete the objects and change the attributes, by replying 'Y'.
- Cancel the queue manager, by replying 'N'.

CSQY341D
Reply Y to continue or N to cancel

Explanation

The installation and customization options chosen for IBM MQ have changed, as indicated in the preceding CSQY340E message.

System action

The queue manager waits for the operator's reply

System programmer response

See message CSQY340E.

CSQY342I
Deleting objects and changing attributes - restricted functionality

Explanation

This message is sent if the operator answers 'Y' to message CSQY341D.

System action

The queue manager deletes the objects and changes the attributes that are not allowed with the restricted functionality.

CSQY343I
Queue manager terminating - restricted functionality not accepted

Explanation

This message is sent if the operator answers 'N' to message CSQY341D.

System action

The queue manager does not start.

z/OS *Advanced Message Security (CSQ0...)*

CSQ0101E

csect-name Internal message protection error, reason *reason*, diagnostics: *value1*,*value2*

Severity
8

Explanation

An internal error occurred during message protection processing.

System action

For a put operation, the message is rejected.

For a get operation, the message is moved to the error queue, **SYSTEM.PROTECTION.ERROR.QUEUE**.

System programmer response

You should check that the message has valid IBM MQ headers, if not, contact IBM Support.

CSQ0105E

csect-name Internal error occurred, reason *reason*, diagnostics: *value*

Severity
8

Explanation

An internal error occurred during message protection processing.

System action

The message queue interface (MQI) call fails.

System programmer response

Contact IBM Support.

CSQ0109E

csect-name SDUMP failed, return code *rc*, reason *reason*

Severity
8

Explanation

An attempt to issue an SDUMP during abend processing failed.

System action

SDUMP diagnostics are not generated.

System programmer response

Review the return code and reason in conjunction with SDUMP documentation to resolve the problem.

CSQ0110I

csect-name AMS abend *abend*, reason *reason*

Severity
8

Explanation

An abend has occurred during message protection processing of type *abend* for reason *reason*.

System action

The message queue interface (MQI) call fails and the IBM MQ subsystem might terminate.

System programmer response

Use the abend and reason code information to resolve the issue.

If the problem cannot be resolved contact your IBM support center.

CSQ0111I

csect-name Module offset *offset*, level *level*

Severity
0

Explanation

The *module* and *level* are reported for diagnostic purposes following an abend during message protection processing.

System action

Processing continues.

System programmer response

None.

CSQ0112I

csect-name PSW SDWAEC1, *SDWAEC1*, SDWAAEC1, *SDWAAEC1*

Severity
0

Explanation

Relevant Program Status Word (PSW) fields are reported following an abend during message protection processing.

System action

Processing continues.

System programmer response

None.

CSQ0113I

csect-name CAB *value1*, *value2*, *value3*, *value4*

Severity
0

Explanation

Relevant internal fields are reported following an abend during message protection processing.

System action

Processing continues.

System programmer response

None.

CSQ0114I

csect-name R0-R3 *gpr0*, *gpr1*, *gpr2*, *gpr3*

Severity
0

Explanation

General purpose registers 0 through 3 are reported following an abend during message protection processing.

System action

Processing continues.

System programmer response

None.

CSQ0115I

csect-name R4-R7 *gpr4*, *gpr5*, *gpr6*, *gpr7*

Severity
0

Explanation

General purpose registers 4 through 7 are reported following an abend during message protection processing.

System action

Processing continues.

System programmer response

None.

CSQ0116I

csect-name R8-R11 *gpr8*, *gpr9*, *gpr10*, *gpr11*

Severity
0

Explanation

General purpose registers 8 through 11 are reported following an abend during message protection processing.

System action

Processing continues.

System programmer response

None.

CSQ0117I

csect-name R12-R15 *gpr12*, *gpr13*, *gpr14*, *gpr15*

Severity
0

Explanation

General purpose registers 12 through 15 are reported following an abend during message protection processing.

System action

Processing continues.

System programmer response

None.

CSQ0118I

csect-name A0-A3 *ar0*, *ar1*, *ar2*, *ar3*

Severity
0

Explanation

Access registers 0 through 3 are reported following an abend during message protection processing.

System action

Processing continues.

System programmer response

None.

CSQ0119I

csect-name A4-A7 *ar4*, *ar5*, *ar6*, *ar7*

Severity
0

Explanation

Access registers 4 through 7 are reported following an abend during message protection processing.

System action

Processing continues.

System programmer response

None.

CSQ0120I

csect-name A8-A11 *ar8*, *ar9*, *ar10*, *ar11*

Severity
0

Explanation

Access registers 8 through 11 are reported following an abend during message protection processing.

System action

Processing continues.

System programmer response

None.

CSQ0121I

csect-name A12-A15 *ar12*, *ar13*, *ar14*, *ar15*

Severity
0

Explanation

Access registers 12 through 15 are reported following an abend during message protection processing.

System action

Processing continues.

System programmer response

None.

CSQ0137I

csect-name SDUMP not taken, suppressed by DAE

Severity
0

Explanation

An SDUMP was suppressed due to Dump Analysis and Elimination (DAE).

System action

Processing continues.

System programmer response

None.

CSQ0151E

csect-name Failed to allocate storage

Severity
8

Explanation

An attempt to allocate storage during message protection processing failed.

System action

The message queue interface (MQI) call fails.

System programmer response

Increase the amount of storage available.

CSQ0174E

csect-name Failed to load module *module*, return code *abncode*, reason *rsncode*

Severity
8

Explanation

An attempt to load a module into storage failed.

System action

The IBM MQ subsystem fails to start.

System programmer response

Use the abend and reason code in conjunction with documentation for the **LOAD** macro to resolve the problem.

CSQ0175E

csect-name Failed to delete module *module*, return code *rc*

Severity
8

Explanation

An attempt to delete a loaded module failed.

System action

The module remains loaded.

System programmer response

Use the return code in conjunction with documentation for the **DELETE** macro to resolve the problem.

CSQ0201E

csect-name Message table not available

Severity
8

Explanation

An attempt to load the message protection component message file failed.

System action

The IBM MQ subsystem fails to start.

System programmer response

Verify that the IBM MQ subsystem has been installed correctly. If the problem persists contact IBM Support.

CSQ0204I

csect-name AMS is using *use-size* MB of local storage, *free-size* MB free

Severity
0

Explanation

The amount of storage currently used for message protection services is currently *use-size* MB, and a further *free-size* remains free.

System action

Processing continues.

System programmer response

None.

CSQ0209E

csect-name Message for *qname* sent to error queue, MQRC=*mqrc (mqrc-text)*

Severity
4

Explanation

During get processing a protected message on queue *qname* could not be processed for reason *mqrc*, and has been sent to the error queue, **SYSTEM.PROTECTION.ERROR.QUEUE** (*mqrc-text* provides the MQRC in textual form).

System action

The message is placed on the error queue and an error is returned to the requesting application.

System programmer response

Examine the message on the error queue and the reason code to determine why the message could not be processed.

You should check the sender and receiver policies. For example, when setting the policy:

- Specify the State or Province using ST=
- The following special characters need escape characters:

```
, (comma)
+ (plus)
" (double quote)
\ (backslash)
< (less than)
> (greater than)
; (semicolon)
```

- If the Distinguished Name contains embedded blanks, you should enclose the DN in double quotation marks.

CSQ0210E

csect-name Failed to redirect message to error queue, MQRC=*mqrc (mqrc-text)*

Severity
8

Explanation

During get procesing, message protection processing failed for reason *mqrc* (*mqrc-text* provides the MQRC in textual form). An attempt to put the message to the error queue, **SYSTEM.PROTECTION.ERROR.QUEUE**, failed.

System action

The get request fails and the message remains on the target queue.

System programmer response

Examine the message on the target queue and the reason code to determine why the message could not be processed or placed on the error queue.

Check the queue manager and Advanced Message Security task error logs for error messages relating to the failure to put the message to the error queue.

CSQ0213E

csect-name Internal queue close failed MQCC=*mqcc* MQRC=*mqrc (mqrc-text)*

Severity
8

Explanation

During open processing for a protected queue protection initialization failed and an attempt to internally close the queue failed.

System action

The open request fails.

System programmer response

Examine the completion and reason codes to determine the cause of the failure.

CSQ0214E

csect-name Message protection initialization failed, return code *rc*, reason *reason*

Severity
8

Explanation

During open processing for a protected queue protection initialization failed.

System action

The open request fails.

System programmer response

Examine the completion and reason codes to determine the cause of the failure. For more information, see Messages and codes in the *z/OS Cryptographic Services System SSL Programming* documentation.

CSQ0215E

csect-name Message protection failed, return code *rc*, reason *reason*

Severity
8

Explanation

An attempt to protect a message failed during put processing.

System action

The message is not put to the queue.

System programmer response

Examine the completion and reason codes to determine the cause of the failure. For more information, see Messages and codes in the *z/OS Cryptographic Services System SSL Programming* documentation.

CSQ0216E

csect-name Message unprotection failed, return code *rc*, reason *reason*

Severity
8

Explanation

An attempt to process a protected message during get processing failed.

System action

If the operation that failed was a destructive get, the message is moved to the error queue, **SYSTEM.PROTECTION.ERROR.QUEUE**, if possible.

System programmer response

Examine the completion and reason codes to determine the cause of the failure. For more information, see Messages and codes in the *z/OS Cryptographic Services System SSL Programming* documentation.

CSQ0217E

csect-name Failed to process object *'objname'*

Severity
8

Explanation

An attempt to initialize, protect, or process a protected message failed for the object named by *objname*.

System action

The open, get or put request fails.

System programmer response

Examine preceding or subsequent console messages for more information.

CSQ0218E

csect-name Privacy policy for *qname* invalid. No recipients

Severity
8

Explanation

During open or put1 processing, a privacy policy was stipulated for the object *qname*, but the policy failed to identify any recipients.

System action

The open or put1 request fails.

System programmer response

Modify or delete the protection policy for the object *qname*.

CSQ0219E

csect-name Message verification error for *qname*

Severity
8

Explanation

During put or get processing an attempt to process a message failed due to unexpected header values or offsets.

System action

The put or get operation fails.

For get processing the message is moved to the error queue, **SYSTEM.PROTECTION.ERROR.QUEUE**.

System programmer response

Examine the failing message to determine the cause of the problem.

CSQ0220E

csect-name Encryption strength not available

Severity
8

Explanation

During get processing a protected message did not specify an encryption strength.

System action

The get request fails and the message is moved to the error queue, **SYSTEM.PROTECTION.ERROR.QUEUE**.

System programmer response

Examine the message on the error queue to determine its origin and why it is not correctly protected.

CSQ0221E

csect-name Message encryption strength *encstr* not valid

Severity
8

Explanation

During get processing a protected message did not have a recognized encryption strength.

System action

The get request fails and the message is moved to the error queue, **SYSTEM.PROTECTION.ERROR.QUEUE**.

System programmer response

Examine the message on the error queue to determine its origin and why it does not have a valid encryption strength.

Some encryption algorithms are supported on some platforms, however, not on others.

CSQ0222E

csect-name Message encryption strength *encstr* inconsistent with policy

Severity
8

Explanation

During get processing a protected message did not use an encryption algorithm that matches the expected encryption strength.

System action

The get request fails and the message is moved to the error queue, **SYSTEM.PROTECTION.ERROR.QUEUE**.

System programmer response

Examine the message on the error queue to determine its origin and why the encryption algorithm does not match the expected encryption strength.

Some encryption algorithms are supported on some platforms, however, not on others.

CSQ0223E

csect-name Message size *m-size* inconsistent with header size *h-size* or original size *o-size*

Severity
8

Explanation

During get processing a protected message was found to have a header or overall message size that did not match the original unprotected message.

System action

The get request fails and the message is moved to the error queue, **SYSTEM.PROTECTION.ERROR.QUEUE**.

System programmer response

Examine the message on the error queue to determine why its lengths are inconsistent with the original unprotected message.

CSQ0224E

csect-name Message buffer length of *m-size* too small

Severity
8

Explanation

During get processing a protected message was of insufficient length to contain a standard protection header and could not be processed.

System action

The get request fails and the message is moved to the error queue, **SYSTEM.PROTECTION.ERROR.QUEUE**.

System programmer response

Examine the message on the error queue to determine why it is of insufficient length.

CSQ0225E

csect-name Message header not acceptable, structure identifier is *'strucid'*

Severity
8

Explanation

During get processing a protected message did not have the expected protection header eye-catcher, but instead had *strucid*.

System action

The get request fails and the message is moved to the error queue, **SYSTEM.PROTECTION.ERROR.QUEUE**.

System programmer response

Examine the message on the error queue to determine why it has an invalid protection header.

CSQ0226E

csect-name Header version not supported

Severity
8

Explanation

During get processing a protected message did not have the expected protection header version.

System action

The get request fails and the message is moved to the error queue, **SYSTEM.PROTECTION.ERROR.QUEUE**.

System programmer response

Examine the message on the error queue to determine why it has an invalid protection header.

CSQ0227E

csect-name Message signature algorithm *sig-alg* not valid

Severity
8

Explanation

During get processing a protected message did not have a recognized signature algorithm.

System action

The get request fails and the message is moved to the error queue, **SYSTEM.PROTECTION.ERROR.QUEUE**.

System programmer response

Examine the message on the error queue to determine its origin and why it does not have a valid signature algorithm. Some signature algorithms are supported on some platforms, however, not on others.

CSQ0228E

csect-name Message signature algorithm *sig-alg* inconsistent with policy

Severity
8

Explanation

During get processing a protected message did not use a signature algorithm that matches the expected signature strength.

System action

The get request fails and the message is moved to the error queue, **SYSTEM.PROTECTION.ERROR.QUEUE**.

System programmer response

Examine the message on the error queue to determine its origin and why the signature algorithm does not match the expected signature strength. Some signature algorithms are supported on some platforms, however, not on others.

CSQ0229E

csect-name Unable to verify sender distinguished name

Severity
8

Explanation

During get processing the distinguished name of the message sender was not present to be verified.

System action

The get request fails and the message is moved to the error queue, **SYSTEM.PROTECTION.ERROR.QUEUE**.

System programmer response

Examine the message on the error queue to determine why it does not have a distinguished name for the sender of the message.

CSQ0230E

csect-name Structure identifier *strucid* invalid for format name *format*

Severity
8

Explanation

During message protection processing a message header did not have the expected value for the message format indicated.

System action

The MQI call fails. For a get request the message is moved to the error queue, **SYSTEM.PROTECTION.ERROR.QUEUE**.

System programmer response

Examine the failing message formats and headers to determine the cause of the problem.

CSQ0231E

csect-name Unrecognized version *version* for structure *strucid*, format name *format*

Severity
8

Explanation

During message protection processing a message header version did not have the expected value for the message format and header indicated.

System action

The MQI call fails. For a get request the message is moved to the error queue, **SYSTEM.PROTECTION.ERROR.QUEUE**.

System programmer response

Examine the failing message formats and headers to determine the cause of the problem.

CSQ0232E

csect-name Buffer length insufficient for format name *format*

Severity
8

Explanation

During message protection processing a message length was insufficient to account for the length of a header indicated by the message format.

System action

The MQI call fails. For a get request the message is moved to the error queue, **SYSTEM.PROTECTION.ERROR.QUEUE**.

System programmer response

Examine the failing message formats and headers to determine the cause of the problem.

CSQ0233E

csect-name Message *msg-size* of size does not match original size of *orig-size*

Severity
8

Explanation

During get processing a protected message length does not resolve to the original length of the message before it was protected.

System action

The get request fails and the message is moved to the error queue, **SYSTEM.PROTECTION.ERROR.QUEUE**.

System programmer response

Examine the message on the error queue to determine why it has a different length than expected.

CSQ0234I

csect-name Policy for *queue-name* not defined, message protection will not be removed

Severity
0

Explanation

SPLPROT(REMOVE) was specified on a channel with CHLTYPE SDR or SVR, but an AMS policy has not been defined for the transmission queue.

The channel will start up, but protection will not be removed from messages retrieved from the transmission queue.

System action

Processing continues.

System programmer response

If message protection is to be removed, define an AMS policy for the transmission queue, refresh AMS policies, and stop and restart the channel.

CSQ0240E

csect-name No storage for error queue processing for *qname*

Severity
8

Explanation

During get processing a message that failed protection processing could not be put to the error queue, **SYSTEM.PROTECTION.ERROR.QUEUE**, due to insufficient storage.

System action

The get request fails and the message remains on the queue *qname*.

System programmer response

Determine the cause of storage shortage and retry the get operation.

CSQ0400I

CSQ0UTIL IBM MQ AMS for z/OS *vrm*

Severity
0

Explanation

The Advanced Message Security policy utility, CSQ0UTIL, is starting for utility version *v*, release *r*, and modlevel *m*.

System action

Processing continues.

System programmer response

None.

CSQ0401I

Queue Manager Protection Policy Utility

Severity
0

Explanation

The Advanced Message Security policy utility, CSQ0UTIL, has started.

System action

Processing continues.

System programmer response

None.

CSQ0402I

Command Name: *command*

Severity
0

Explanation

The Advanced Message Security policy utility is processing the policy command *command*.

System action

Processing continues.

System programmer response

None.

CSQ0403I

Arguments: args

Severity
0

Explanation

The Advanced Message Security policy utility is processing arguments *args* for the current policy command.

System action

Processing continues.

System programmer response

None.

CSQ0404E

Insufficient storage available to perform command

Severity
8

Explanation

The Advanced Message Security policy utility could not allocate storage to process the input command.

System action

The Advanced Message Security policy utility terminates without executing the current input command.

System programmer response

Determine why there is insufficient storage for the policy utility then rerun the utility when the problem has been resolved.

CSQ0405E

An error occurred running command *cmd-number* on line *line number*

Severity
8

Explanation

The Advanced Message Security policy utility encountered an error during the processing of command *cmd-number* at line *line-number* of the input.

System action

The Advanced Message Security policy utility processing fails for the command identified by *cmd-number* at line *line-number*.

System programmer response

Examine the failing command and related messages to determine the cause of the failure.

CSQ0406E

Invalid command found on line *line-number*. Valid commands are SETMQSPL and DSPMQSPL

Severity
8

Explanation

The Advanced Message Security policy utility did not recognize the input command at line *line-number* of the input.

System action

The Advanced Message Security policy utility does not process the command at line *line-number*.

System programmer response

Change the input command to either **SETMQSPL** or **DSPMQSPL**.

CSQ0407E

Quoted string on line *line-number* does not have a terminating quote

Severity
8

Explanation

The Advanced Message Security policy utility could not align matching quotes when processing the input command at line *line-number*.

System action

The Advanced Message Security policy utility does not process the command at line *line-number*.

System programmer response

Change the command at line *line-number* to use consistent and matching quotes in the arguments.

CSQ0408I

cmd-count policy commands have been completed successfully

Severity
0

Explanation

The Advanced Message Security policy utility has successfully processed *cmd-count* commands.

System action

Processing continues.

System programmer response

None.

CSQ0409I

Reached end of input, *num-line* lines read

Severity
0

Explanation

The Advanced Message Security policy utility reached end of input after *num-line* lines.

System action

Processing continues.

System programmer response

None.

CSQ0410E

Error opening SYSIN data set

Severity
8

Explanation

The Advanced Message Security policy utility could not open the standard input (SYSIN DD) to read input commands.

System action

No commands are processed.

System programmer response

Determine why the SYSIN DD is unavailable and resolve the problem, then rerun the policy utility.

CSQ0411E

Unexpected internal error

Severity
8

Explanation

The Advanced Message Security policy utility did not recognize the input command.

System action

The input command is not processed.

System programmer response

Examine the command input and verify that the input expresses a valid command with valid parameters.

CSQ0412I

Policy name: *policy-name*

Severity
0

Explanation

The Advanced Message Security policy utility is displaying information about the policy identified by *policy-name*.

System action

Processing continues.

System programmer response

None.

CSQ0413I

Encryption algorithm: *enc-alg*

Severity
0

Explanation

The Advanced Message Security policy utility is displaying the encryption algorithm *enc-alg* for a given policy.

System action

Processing continues.

System programmer response

None.

CSQ0414I

Recipient DNs: *recipient-dns*

Severity
0

Explanation

The Advanced Message Security policy utility is displaying the recipient distinguished names, *recipient-dns*, for a given policy.

System action

Processing continues.

System programmer response

None.

CSQ0415I

Signature algorithm: *sig-alg*

Severity
0

Explanation

The Advanced Message Security policy utility is displaying the signature algorithm, *sig-alg*, for a given policy.

System action

Processing continues.

System programmer response

None.

CSQ0416I

Signer DNs: *signer-dns*

Severity
0

Explanation

The Advanced Message Security policy utility is displaying the signer distinguished names, *signer-dns*, for a given policy.

System action

Processing continues.

System programmer response

None.

CSQ0417I

Quality of protection: *qop*

Severity
0

Explanation

The Advanced Message Security policy utility is displaying the quality of protection, *qop*, for a given policy.

System action

Processing continues.

System programmer response

None.

CSQ0418I

Toleration: *toleration-flag*

Severity
0

Explanation

The Advanced Message Security policy utility is displaying the toleration flag, *toleration-flag*, for a given policy.

System action

Processing continues.

System programmer response

None.

CSQ0447E

Failed to open EXPORT DD, exporting to STDOUT

Severity
8

Explanation

The Advanced Message Security policy utility could not open the EXPORT DD to process a -export request.

System action

The policy export is sent to STDOUT.

System programmer response

Determine why the EXPORT DD is unavailable and resolve the problem, then rerun the policy utility.

CSQ0448E

Command failed

Severity
8

Explanation

The Advanced Message Security policy utility failed to successfully process an input command.

System action

The current command is not processed and the Advanced Message Security policy utility attempts to process the next input command, if any.

System programmer response

Determine the reason for the failing command by examining related messages, and correct the failing input command.

CSQ0449I

Command successful

Severity
0

Explanation

The Advanced Message Security policy utility successfully processed an input command.

System action

Processing continues.

System programmer response

None.

CSQ0450E

Syntax error. Usage: setmqspl -m (qm) -p (policy) -s (sigalg) -a (signer DN) -e (encalg) -r (receiver DN)

Severity
8

Explanation

The Advanced Message Security policy utility failed to interpret a command due to bad command syntax.

System action

The current command is not processed and the Advanced Message Security policy utility attempts to process the next input command, if any.

System programmer response

Correct the syntax of the failing command then retry.

CSQ0451E

Invalid queue manager name: *qmgr-name*

Severity
8

Explanation

The Advanced Message Security policy utility encountered an invalid queue manager name value, *qmgr-name*, when processing an input command.

System action

The current command is not processed and the Advanced Message Security policy utility attempts to process the next input command, if any.

System programmer response

Correct the queue manager name value in the input command then retry.

CSQ0452E

Invalid policy name: *policy-name*

Severity
8

Explanation

The Advanced Message Security policy utility encountered an invalid policy name when processing an input command.

System action

The current command is not processed and the Advanced Message Security policy utility attempts to process the next input command, if any.

System programmer response

Correct the policy name value in the input command then retry.

CSQ0453E

Invalid encryption algorithm

Severity
8

Explanation

The Advanced Message Security policy utility encountered an invalid encryption algorithm when processing an input command.

System action

The current command is not processed and the Advanced Message Security policy utility attempts to process the next input command, if any.

System programmer response

Correct the encryption algorithm value in the input command then retry.

CSQ0454E

Invalid signature algorithm

Severity
8

Explanation

The Advanced Message Security policy utility encountered an invalid signature algorithm when processing an input command.

System action

The current command is not processed and the Advanced Message Security policy utility attempts to process the next input command, if any.

System programmer response

Correct the signature algorithm value in the input command and retry.

CSQ0455E

Encryption requires the use of a signature algorithm

Severity
8

Explanation

The Advanced Message Security policy utility encountered an invalid command that identified an encryption algorithm, but did not also identify a valid signature algorithm.

System action

The current command is not processed and the Advanced Message Security policy utility attempts to process the next input command, if any.

System programmer response

Provide both a valid encryption algorithm and a valid signature algorithm when defining privacy protection policies.

CSQ0456E

Encryption requires a receiver DN to be specified (-r)

Severity
8

Explanation

The Advanced Message Security policy utility encountered an invalid command that identified an encryption algorithm, but did not also identify at least one recipient DN via the -r parameter.

System action

The current command is not processed and the Advanced Message Security policy utility attempts to process the next input command, if any.

System programmer response

Provide both an encryption algorithm and at least one recipient DN when defining privacy protection policies.

CSQ0457E

Invalid receiver DN specified: *receiver-dn*

Severity
8

Explanation

The Advanced Message Security policy utility encountered an invalid receiver distinguished name, *receiver-dn*, when processing an input command.

System action

The current command is not processed and the Advanced Message Security policy utility attempts to process the next input command, if any.

System programmer response

Correct the receiver distinguished name in the input command then retry.

CSQ0458E

Receiver DN is specified while no encryption is enabled

Severity
8

Explanation

The Advanced Message Security policy utility encountered an invalid command that identified at least one recipient DN, however, did not also identify an encryption algorithm when processing the -e parameter.

System action

The current command is not processed and the Advanced Message Security policy utility attempts to process the next input command, if any.

System programmer response

Provide both an encryption algorithm and at least one recipient DN when defining privacy protection policies.

CSQ0459E

Invalid signer DN specified: *signer-dn*

Severity
8

Explanation

The Advanced Message Security policy utility encountered an invalid signer distinguished name, *signer-dn*, when processing an input command.

System action

The current command is not processed and the Advanced Message Security policy utility attempts to process the next input command, if any.

System programmer response

Correct the signer distinguished name value in the input command then retry.

CSQ0460E

Signer DN is specified while no signing is enabled

Severity
8

Explanation

The Advanced Message Security policy utility encountered an invalid command that identified at least one signer DN using the -a parameter, however, did not also identify a signature algorithm using the -s parameter.

System action

The current command is not processed and the Advanced Message Security policy utility attempts to process the next input command, if any.

System programmer response

Provide both an encryption algorithm and at least one recipient DN when defining privacy protection policies.

CSQ0461E

Queue **SYSTEM.PROTECTION.POLICY.QUEUE** unavailable, MQCC=*mqcc* MQRC=*mqrc (mqrc-text)*

Severity
8

Explanation

The Advanced Message Security policy utility could not open the policy queue, **SYSTEM.PROTECTION.POLICY.QUEUE**, due to an error identified by *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form).

System action

Processing ends.

System programmer response

Determine the reason the policy queue is unavailable using the *mqcc* and *mqrc*, then resolve the problem.

CSQ0462E

Failed to retrieve protection policy, MQCC=*mqcc* MQRC=*mqrc (mqrc-text)*

Severity
8

Explanation

The Advanced Message Security policy utility could not retrieve a policy from the policy queue, **SYSTEM.PROTECTION.POLICY.QUEUE**, due to an error identified by *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form).

System action

The current command is not processed and the Advanced Message Security policy utility attempts to process the next input command, if any.

System programmer response

Determine the reason the policy could not be retrieved from the policy queue, then resolve the problem.

CSQ0463E

Policy update failed due to concurrent update, MQCC=*mqcc* MQRC=*mqrc (mqrc-text)*

Severity
8

Explanation

The Advanced Message Security policy utility detected that a policy was changed by another process when it was trying to update or create the same policy, due to an error identified by *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form).

System action

The current command is not processed and the Advanced Message Security policy utility attempts to process the next input command, if any.

System programmer response

Verify the policy is correct, then update the policy again if necessary.

CSQ0464E

Policy definition not found, MQCC=*mqcc* MQRC=*mqrc (mqrc-text)*

Severity
8

Explanation

The Advanced Message Security policy utility could not find a policy on the policy queue, **SYSTEM.PROTECTION.POLICY.QUEUE**, when a policy was expected, due to an error identified by *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form).

System action

The current command is not processed and the Advanced Message Security policy utility attempts to process the next input command, if any.

System programmer response

Determine the reason the policy could not be retrieved from the policy queue, then resolve the problem.

CSQ0465E

An unexpected error occurred, MQCC=*mqcc* MQRC=*mqrc (mqrc-text)*

Severity
8

Explanation

The Advanced Message Security policy utility encountered an unexpected MQI error when processing an input command, identified by *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form).

System action

The current command is not processed and the Advanced Message Security policy utility attempts to process the next input command, if any.

System programmer response

Determine the reason the for the MQI error, then resolve the problem.

CSQ0466E

Invalid value specified for toleration flag, specify one of (0, 1)

Severity
8

Explanation

The Advanced Message Security policy utility encountered an invalid value for the toleration parameter.

System action

The current command is not processed and the Advanced Message Security policy utility attempts to process the next input command, if any.

System programmer response

Use a value of 0 (false) or 1 (true) for the toleration parameter when creating or modifying a policy.

CSQ0467E

Failed to connect to the queue manager, MQCC=*mqcc* MQRC=*mqrc (mqrc-text)*

Severity
8

Explanation

The Advanced Message Security policy utility could not connect to the input queue manager to process further input commands, due to an error identified by *mqcc* and *mqrc* (*mqrc-text* provides the MQRC in textual form).

System action

Processing ends.

System programmer response

Determine the reason the queue manager is unavailable, then resolve the problem.

CSQ0468I

No policies found

Severity
0

Explanation

The Advanced Message Security policy utility found no policies matching the specified parameters.

System action

Processing continues.

System programmer response

None.

CSQ0469E

Invalid value specified for key reuse argument

Severity
8

Explanation

The Advanced Message Security policy utility encountered an invalid value for the key reuse parameter.

System action

The current command is not processed and the Advanced Message Security policy utility attempts to process the next input command, if any.

System programmer response

Use a valid value for the key reuse parameter when creating or modifying a policy.

CSQ0470E

Syntax error. Usage: dspmqspl -m (qm) -p (policy) -export

Severity
8

Explanation

The Advanced Message Security policy utility failed to interpret a command due to incorrect syntax.

System action

The current command is not processed and the Advanced Message Security policy utility attempts to process the next input command, if any.

System programmer response

Correct the syntax of the failing command then retry.

CSQ0471E

Key reuse not valid for policy

Severity
8

Explanation

The Advanced Message Security policy utility encountered an invalid command that specified a non-zero value for the key reuse parameter for a policy that does not allow symmetric key reuse.

System action

The current command is not processed and the Advanced Message Security policy utility attempts to process the next input command, if any.

System programmer response

Either specify a value of `0` for the key reuse parameter (key reuse disabled), or change the policy to use a quality of protection that allows symmetric key reuse, for example, confidentiality.

CSQ0499I

CSQ0UTIL Utility completed return code=*retcode*

Severity
0

Explanation

The Advanced Message Security policy utility, CSQ0UTIL, has completed with return code *retcode*.

System action

Processing continues.

System programmer response

If the utility did not complete successfully refer to other messages in the output to determine the cause of any errors.

CSQ0501I

csect-name SMF recording enabled for record type *record-type*

Severity
0

Explanation

Advanced Message Security has enabled SMF record generation for record type *record-type*.

System action

Processing continues.

System programmer response

None.

CSQ0502I

csect-name SMF recording disabled

Severity
0

Explanation

Advanced Message Security has disabled SMF record generation.

System action

Processing continues.

System programmer response

None.

CSQ0503I

csect-name SMF record write failed, return code *retcode*

Severity
8

Explanation

An attempt the generate an SMF audit record using SMFEWTM failed during message protection processing with return code *retcode*.

System action

The SMF record is not generated.

System programmer response

Examine the *retcode* and documentation for the SMFEWTM macro to determine the cause of the failure.

CSQ0600I

csect-name IBM MQ AMS for z/OS , *version*, *service-level*

Severity
0

Explanation

The Advanced Message Security task is running at version *version* and service level *service-level*.

System action

Processing continues.

System programmer response

None.

CSQ0601I

csect-name Environment variable *varname* has an invalid value, using default *'value'*

Severity
8

Explanation

A Advanced Message Security environment variable, *varname*, was set to an invalid value, resulting in the use of a default value, *value*, for the variable.

System action

Processing continues with the default value for the named environment variable.

System programmer response

Change the environment variable assignment to a valid value if the default is not acceptable.

CSQ0602I

csect-name AMS initialization complete

Severity
0

Explanation

The Advanced Message Security task initialization is complete.

System action

Processing continues.

System programmer response

None.

CSQ0603I

csect-name AMS shutdown requested

Severity
0

Explanation

The Advanced Message Security task has received a shutdown request.

System action

Processing continues.

System programmer response

None.

CSQ0604I

csect-name LOG option processed: *log-option*

Severity
0

Explanation

The Advanced Message Security task processed a LOG command for log option *log-option*.

System action

Processing continues with the new log option.

System programmer response

None.

CSQ0605E

csect-name Incorrect LOG option specified

Severity
8

Explanation

An attempt by the Advanced Message Security task to process a LOG command failed due to an invalid log option.

System action

The LOG command does not take effect.

System programmer response

Correct the LOG option and retry the LOG command.

CSQ0606E

csect-name Unrecognized command: specify **DISPLAY**, **REFRESH**, **LOG**, **SMFTYPE**, **SMFAUDIT** or **STOP**

Severity
8

Explanation

An attempt by the Advanced Message Security task to process a command failed because the command was not recognized.

System action

The command is not executed.

System programmer response

Select a valid command and retry.

Valid commands include **DISPLAY**, **REFRESH**, **LOG**, **SMFTYPE**, **SMFAUDIT** and **STOP**.

CSQ0607E

csect-name Insufficient storage available

Severity
8

Explanation

The Advanced Message Security task failed to allocate storage.

System action

The function being performed by the Advanced Message Security task fails.

System programmer response

Determine the reason there is insufficient storage and correct or increase, as appropriate.

CSQ0608E

csect-name Failed to load policy configuration, MQRC=*mqrc (mqrc-text)*

Severity
8

Explanation

The Advanced Message Security task failed to load the policy configuration for reason *mqrc* (*mqrc-text* provides the MQRC in textual form).

System action

The Advanced Message Security task cannot start.

System programmer response

Use the reason code, *mqrc*, to determine why the policy configuration could not be loaded from the policy queue, **SYSTEM.PROTECTION.POLICY.QUEUE**.

If the queue manager cannot start, you can define the queue in the CSQINP2 concatenation. Definitions for queues required by Advanced Message Security are provided in SCSQPROC member CSQ4INSM.

CSQ0609I

csect-name AMS for z/OS starting, version *version*, level *service-level*

Severity
0

Explanation

Advanced Message Security task has started for product version *version* and service level *service-level*.

System action

Processing continues.

System programmer response

None.

CSQ0610E

csect-name Failed to start policy subtask, error *errcode*, reason *reason*

Severity
8

Explanation

An attempt by the Advanced Message Security task to start the policy configuration subtask failed with errno *errcode* and errno2 *reason*.

System action

The Advanced Message Security task cannot start.

System programmer response

Use the error and reason codes to determine why the policy configuration subtask could not be started, then take corrective action.

CSQ0611E

csect-name Failed to make AMS address space non-swapable, error *errcode*

Severity
8

Explanation

An attempt by the Advanced Message Security task to make itself non-swapable failed with error code *errcode*.

System action

The Advanced Message Security task cannot start.

System programmer response

The error identified by *errcode* is likely to be the return code from the SYSEVENT macro. Use macro documentation to determine the cause of the SYSEVENT failure.

CSQ0612E

csect-name System function *'function'* failed

Severity
8

Explanation

An attempt by the Advanced Message Security task to use a run-time call, *function*, failed.

System action

The Advanced Message Security task cannot continue to process the service it was providing at the time of the failure.

System programmer response

This message is associated with other messages that are generated at the time of failure. Examine these messages for more information, including error codes that might identify the cause of the failure.

CSQ0613E

csect-name AMS initialization error *errno*, reason *errno2*

Severity
8

Explanation

The Advanced Message Security task failed to initialize due to a run-time call failure.

System action

The Advanced Message Security task cannot start.

System programmer response

This message is associated with other messages that are generated at the time of failure. Examine these messages for more information, and use the error codes to determine the cause of the failure.

CSQ0614E

csect-name AMS termination error *errno*, reason *reason*

Severity
8

Explanation

The Advanced Message Security task failed during termination due to a run-time call failure.

System action

The Advanced Message Security task termination continues.

System programmer response

This message is associated with other messages that are generated at the time of failure. Examine these messages for more information, and use the error codes to determine the cause of the failure.

CSQ0615E

csect-name AMS post/wait request failed, reason *reason*

Severity
8

Explanation

An attempt by the Advanced Message Security task to issue a post or wait request failed for reason *reason*.

System action

The Advanced Message Security task cannot continue to process the service it was providing at the time of the failure.

System programmer response

The error identified by *reason* is likely to be the return code from the POST or WAIT macro. Use macro documentation to determine the cause of the failure.

CSQ0616E

csect-name AMS runtime environment initialization failed

Severity
8

Explanation

The Advanced Message Security task failed to initialize.

System action

The Advanced Message Security task cannot start.

System programmer response

Examine associated messages for more information about the failure, then take corrective action.

CSQ0617E

csect-name AMS already active

Severity
8

Explanation

An attempt to start the Advanced Message Security task failed because it was already running.

System action

The Advanced Message Security task cannot start while it is already running.

System programmer response

None.

CSQ0618E

csect-name AMS initialization failed, program not APF authorized

Severity
8

Explanation

An attempt to start the Advanced Message Security task failed because the module, CSQ0DSRV, is not APF authorized.

System action

The Advanced Message Security task cannot start.

System programmer response

Ensure that the AMS task module is APF authorized and retry.

CSQ0619I
csect-name AMSPROD=*prod-value*, recording product usage for *product-name* product ID *product-id*

Explanation

This message is issued when AMS starts, and if SMF 89 product usage records are to be recorded by AMS:

prod-value is the value defined for AMSPROD, and is allowed to be blank (see START QMGR for information on how to set `prod-value`.
product-name is the descriptive name of the product
product-id is the product ID that is to be used in the SMF 89 data

For example:

- AMSPROD=ADVANCEDVUE, recording product usage for IBM MQ Advanced for z/OS Value Unit Edition product ID 5656-AVA - this indicates that usage data is to be collected for the IBM MQ Advanced for z/OS Value Unit Edition product.

See *z/OS MVS Product Management* for more information on product usage recording.

CSQ0624E

csect-name SMF audit option invalid, defaulting to 'failure'

Severity
8

Explanation

An attempt to process the _AMS_SMF_AUDIT environment variable or an SMFAUDIT command failed because the variable or command value was not recognized.

System action

The variable assignment or command is ignored and the default value 'failure' is used.

System programmer response

Provide a valid variable or command value. Valid values include 'success', 'failure', and 'all'.

CSQ0625E

csect-name SMF record type invalid

Severity
8

Explanation

An attempt to process the _AMS_SMF_TYPE environment variable or an SMFTYPE command failed because the variable or command value was not valid.

System action

The variable assignment or command is ignored.

System programmer response

Provide a valid variable or command value. Valid values include numeric values between 0 and 255 inclusive. The SMFTYPE value represents the SMF record type for SMF record generation. A value of 0 means no SMF record generation is required. The recommended value is 180.

CSQ0626I

csect-name SMF audit type is *audit-type*

Severity
0

Explanation

The Advanced Message Security SMF audit type has been set to *audit-type*.

System action

The new SMF audit type takes effect immediately. If *audit-type* is 'failure', all failing puts/gets to a protected queue are audited. If *audit-type* is 'success', all successful puts/gets to a protected queue are audited. If *audit-type* is 'all', both successful and failing puts/gets to a protected queue are audited.

System programmer response

None.

CSQ0629E

csect-name Unable to create security environment for user *userid*, return code *errno*, reason *errno2*.

Severity
8

Explanation

An attempt by the Advanced Message Security task to create a thread-level security environment using `pthread_security_np()` for user *userid* failed for the reasons indicated by *errno* and *errno2*.

System action

The thread-level security environment is not created, and the AMS function being processed cannot be completed. The MQI call fails.

System programmer response

Examine the *errno* and *errno2* values in conjunction with `pthread_security_np()` documentation to determine the cause of the failure.

CSQ0630E

csect-name Unable to delete security environment, reason *errno*

Severity
8

Explanation

An attempt by the Advanced Message Security task to delete a thread-level security environment using pthread_security_np() failed for the reason indicated by *errno*.

System action

The thread-level security environment is not deleted. AMS processing continues.

System programmer response

Examine the *errno* value in conjunction with pthread_security_np() documentation to determine the cause of the failure.

CSQ0631E

csect-name AMS not started, product is not enabled

Severity
8

Explanation

An attempt by the Advanced Message Security task to register itself using macro IFAEDREG failed.

System action

The Advanced Message Security task cannot start.

System programmer response

Verify that the PARMLIB IFAPRDxx member has been built with the provided AMS product information, then retry.

CSQ0632E

csect-name AMS deregistration failed, reason *reason*

Severity
8

Explanation

An attempt by the Advanced Message Security task to deregister itself using macro IFAEDDRG failed.

System action

The Advanced Message Security task cannot deregister. Processing continues.

System programmer response

Examine the reason returned by the IFAEDDRG macro in conjunction with macro documentation to determine the cause of the failure.

CSQ0633I

csect-name AMS environment variable values:

Severity
0

Explanation

The Advanced Message Security task identifies its environment variables and their values immediately following this message.

System action

Processing continues.

System programmer response

None.

CSQ0634I

csect-namevariable=value

Severity
0

Explanation

During startup, the Advanced Message Security task issues this message to report an environment variable *variable*, and its value *value*.

System action

Processing continues.

System programmer response

None.

CSQ0635I

csect-name POLICY refresh complete

Severity
0

Explanation

The Advanced Message Security task has refreshed its policy configuration in response to a **REFRESH** command.

System action

Processing continues.

System programmer response

None.

CSQ0636I

csect-name POLICY refresh failed

Severity
8

Explanation

An attempt by the Advanced Message Security task to refresh its policy configuration failed.

System action

The policy configuration is not refreshed.

System programmer response

Examine the console for associated error messages to determine the cause of the failure.

CSQ0637I

csect-name KEYRING refresh complete

Severity
0

Explanation

The Advanced Message Security task has refreshed its keyring configuration in response to a **REFRESH** command.

System action

Processing continues.

System programmer response

None.

CSQ0638E

csect-name KEYRING refresh failed, return code *errno*

Severity
8

Explanation

An attempt by the Advanced Message Security task to refresh its keyring configuration failed for the reason indicated by *errno*.

System action

The keyring configuration is not refreshed.

System programmer response

Examine the console for associated error message to determine the cause of the failure. Use the *errno*, which might represent a System SSL gsk_status to further diagnose the problem.

CSQ0639E

csect-name Incorrect **REFRESH** option, specify KEYRING, POLICY or ALL

Severity
8

Explanation

An attempt by the Advanced Message Security task to process a **REFRESH** command failed because the refresh option was not recognized.

System action

The **REFRESH** command is not processed.

System programmer response

Ensure the **REFRESH** option is KEYRING, POLICY or ALL, depending on which option should be refreshed.

CSQ0640E

csect-name AMS not started correctly

Severity
8

Explanation

The Advanced Message Security task has started incorrectly.

System action

The Advanced Message Security task fails to start.

System programmer response

The Advanced Message Security task can only be started internally by IBM MQ.

CSQ0641I

csect-name **REFRESH** command completed successfully

Severity
0

Explanation

The Advanced Message Security task has successfully processed a **REFRESH** command.

System action

Processing continues.

System programmer response

None.

CSQ0642I

*csect-name***REFRESH** command failed

Severity
8

Explanation

The Advanced Message Security task has failed to successfully process a **REFRESH** command.

System action

The requested **REFRESH** command is not processed.

System programmer response

Examine the console for associated error messages to determine the cause of the problem.

CSQ0648E

csect-name Failed to open AMS key ring, reason *gsk-status*

Severity
8

Explanation

An attempt by the Advanced Message Security task to open its keyring failed for the reason indicated by *gsk-status*.

System action

The AMS keyring is not opened, and the AMS task cannot start.

System programmer response

Examine System SSL documentation related to the gsk_open_keyring() call in conjunction with the *gsk-status* code to determine the cause of the failure.

CSQ0649E

csect-name CRL initialization failed

Severity
8

Explanation

An attempt by the Advanced Message Security task to connect to an LDAP server, based on configuration provided in the CRLFILE DD, failed.

System action

The Advanced Message Security task cannot perform Certificate Revocation List (CRL) checking. The behavior of certificate validation is determined by the System SSL environment variable GSK_CRL_SECURITY_LEVEL. See System SSL documentation for more information.

System programmer response

Check the configuration provided via the CRLFILE DD in the AMS started task JCL and verify that the configuration details are correct.

CSQ0651E

csect-name Failed to open CRL LDAP, *ldap-name*

Severity
8

Explanation

An attempt by the Advanced Message Security task to open an LDAP directory, *ldap-name*, failed.

System action

The Advanced Message Security task cannot perform Certificate Revocation List (CRL) checking against the named LDAP directory. The behavior of certificate validation is determined by the System SSL environment variable GSK_CRL_SECURITY_LEVEL. See System SSL documentation for more information.

System programmer response

Check the configuration provided in the CRLFILE DD in the AMS started task JCL and verify that the configuration details are correct. Verify that the failing directory is available.

CSQ0652I

csect-name CRL checking enabled

Severity
0

Explanation

The Advanced Message Security task has successfully enabled Certificate Revocation List (CRL) checking.

System action

Processing continues.

System programmer response

None.

CSQ0653I

csect-name CRL checking disabled

Severity
0

Explanation

The Advanced Message Security task has successfully disabled Certificate Revocation List (CRL) checking.

System action

Processing continues.

System programmer response

None.

CSQ0660E

csect-name Internal version mismatch

Severity
8

Explanation

The Advanced Message Security task has received a request for data protection services with an unrecognized request version value.

System action

The data protection service cannot be provided.

System programmer response

This error implies that a task other than the Advanced Message Security Interceptor is attempting to exploit AMS data protection services. AMS data protection services are only available by using the AMS Interceptor.

CSQ0699I

csect-name AMS shutdown complete

Severity
0

Explanation

The Advanced Message Security task has shutdown.

System action

Processing continues.

System programmer response

None.

CSQ0996I

csect-name char-diag1, char-diag2, char-diag3, char-diag4, hex-diag1, hex-diag2

Severity
0

Explanation

This message is generated when Advanced Message Security is running in DEBUG mode, as directed by IBM Support, and provides character and hexadecimal diagnostic values to aid in problem resolution.

System action

Processing continues.

System programmer response

None.

CSQ0997I

csect-name char-diag1, char-diag2, char-diag3, hex-diag1, hex-diag2, hex-diag3

Severity
0

Explanation

This message is generated when Advanced Message Security is running in DEBUG mode, as directed by IBM Support, and provides character and hexadecimal diagnostic values to aid in problem resolution.

System action

Processing continues.

System programmer response

None.

CSQ0998I

csect-name char-diag1, char-diag2, hex-diag1, hex-diag2, hex-diag3, hex-diag4

Severity
0

Explanation

This message is generated when Advanced Message Security is running in DEBUG mode, as directed by IBM Support, and provides character and hexadecimal diagnostic values to aid in problem resolution.

System action

Processing continues.

System programmer response

None.

CSQ0999I

csect-name char-diag1, hex-diag1, hex-diag2, hex-diag3, hex-diag4, hex-diag5

Severity
0

Explanation

This message is generated when Advanced Message Security is running in DEBUG mode, as directed by IBM Support, and provides character and hexadecimal diagnostic values to aid in problem resolution.

System action

Processing continues.

System programmer response

None.

z/OS *Service facilities messages (CSQ1...)*

The value shown for severity in the service facility messages that follow is the value returned as the job-step condition code from the job-step during which the message is issued. If additional messages having higher severity values are issued during the same job-step, the higher value is reflected as the job-step condition code.

Log services return codes

The return codes set by log services are:

0
Successful completion

4
Exception condition (for example, end of file), not an error.

8
Unsuccessful completion due to parameter errors.

12
Unsuccessful completion. Error encountered during processing of a valid request.

CSQ1000I

csect-name IBM MQ for z/OS V*n*

Severity
0

Explanation

This message is issued as the first part of the header to the report issued by the log print utility program.

CSQ1100I

csect-name LOG PRINT UTILITY - *date time*

Severity
0

Explanation

This message is issued as the second part of the header to the report issued by the log print utility program.

CSQ1101I

csect-name UTILITY PROCESSING COMPLETED, RETURN CODE=*rc*

Severity
0

Explanation

The log print utility completed with the return code *rc* indicated. 0 indicates successful completion.

CSQ1102I

SEARCH CRITERIA

Severity
0

Explanation

The search criteria specified for printing the log follow.

CSQ1105I

LOG PRINT UTILITY SUMMARY - *date time*

Severity
0

Explanation

This is issued as a header to the summary data set written by the log print utility.

CSQ1106I

END OF SUMMARY

Severity
0

Explanation

This marks the end of the summary data set written by the log print utility.

CSQ1110E

LIMIT OF 50 STATEMENTS EXCEEDED

Severity
8

Explanation

The limit of 50 input statements allowed by CSQ1LOGP has been exceeded.

System action

Processing is terminated.

System programmer response

Resubmit the job using no more than 50 statements.

CSQ1111E

LIMIT OF 80 TOKENS EXCEEDED

Severity
8

Explanation

The limit of 80 keywords and corresponding value specifications allowed by CSQ1LOGP has been exceeded. A keyword with its value is considered as two tokens.

System action

Processing is terminated.

System programmer response

Resubmit the job using no more than 80 tokens.

CSQ1112E

TOKEN *xxx*... EXCEEDS 48 CHARACTERS

Severity
8

Explanation

An input statement contains the character string beginning *xxx*. This string is not valid because it exceeds 48 characters in length.

System action

Processing is terminated.

System programmer response

Resubmit the job with a valid token.

CSQ1113E

INVALID SYNTAX FOR KEYWORD *kwd*

Severity
8

Explanation

An input statement contains the keyword *kwd*. The value specified for this keyword is not valid, because it is not of the form *kwd*(*value*).

System action

Processing is terminated.

System programmer response

Resubmit the job with the correct form of the keyword.

CSQ1127E

KEYWORD *kwd* UNKNOWN

Severity
8

Explanation

CSQ1LOGP does not recognize the keyword *kwd*.

System action

Processing is terminated.

System programmer response

Check to make sure all keywords are valid and resubmit the job.

CSQ1128E

END OF LOG RANGE SPECIFIED WITHOUT START

Severity
8

Explanation

You cannot specify the end of a search range (RBAEND or LRSNEND) without specifying a beginning of the search range (RBASTART or LRSNSTART).

System action

Processing is terminated.

System programmer response

Resubmit the job providing an RBASTART or LRSNSTART value to correspond to the RBAEND or LRSNEND value given to specify a valid search range.

CSQ1129E

LIMIT OF 10 *kwd* KEYWORDS EXCEEDED

Severity
8

Explanation

The *kwd* keyword appears too many times in the control statements. The limit is 10.

System action

Processing is terminated.

System programmer response

Resubmit the job providing no more than 10 of these keywords.

CSQ1130E

INVALID VALUE FOR KEYWORD *kwd* NUMBER *n*

Severity
8

Explanation

The value for the *n*th occurrence of keyword *kwd* is invalid because it has invalid characters, it is not one of a list of permitted values, or it is too long.

System action

Processing is terminated.

System programmer response

Resubmit the job providing a correct value specification.

CSQ1131E

INVALID VALUE FOR KEYWORD *kwd*

Severity
8

Explanation

The value for the keyword *kwd* is invalid because it has invalid characters, it is not one of a list of permitted values, or it is too long.

System action

Processing is terminated.

System programmer response

Resubmit the job providing a correct value specification.

CSQ1132E

NO VALUE FOR KEYWORD *kwd* NUMBER *n*

Severity
8

Explanation

The *n*th occurrence of keyword *kwd* is not followed by a value.

System action

Processing is terminated.

System programmer response

Resubmit the job providing a correct value specification.

CSQ1133E

NO VALUE FOR KEYWORD *kwd*

Severity
8

Explanation

The keyword *kwd* is not followed by a value.

System action

Processing is terminated.

System programmer response

Resubmit the job providing a correct value specification.

CSQ1134E

KEYWORD EXTRACT REQUIRES AT LEAST ONE OUTPUT DDNAME

Severity
4

Explanation

The keyword extract requires at least one output DDNAME for log records to be extracted.

System action

Processing continues, however, no log records are extracted .

System programmer response

Remove EXTRACT(YES), or alternatively add a DDNAME from the following list: **CSQBACK**, **CSQCMT**, **CSQBOTH**, **CSQINFLT**, **CSQOBJS**. Resubmit the job. For more information see, The log print utility (CSQ1LOGP).

CSQ1135E

KEYWORD *kwd* SPECIFIED MORE THAN ONCE

Severity
8

Explanation

The keyword *kwd* can only be specified once.

System action

Processing is terminated.

System programmer response

Resubmit the job providing only one of these keywords.

CSQ1137I

FIRST PAGE SET CONTROL RECORD AFTER RESTART = *r-rba*

Severity
0

Explanation

r-rba is the log RBA of a record that serves as an implicit indication that a restart occurred just prior to this point.

System action

Processing continues.

CSQ1138E
kwd1 AND *kwd2* CANNOT BOTH BE SPECIFIED

Explanation

kwd1 and *kwd2* cannot both appear in the control statements.

System action

Processing is terminated.

System programmer response

Correct the control statements and rerun the job.

CSQ1139E

SYSSUMRY DD STATEMENT MISSING

Severity
8

Explanation

You requested the SUMMARY option, but did not include the SYSSUMRY DD statement in your JCL.

System action

Processing terminates.

System programmer response

Resubmit the job with a SYSSUMRY DD statement included in the JCL.

CSQ1145E

CURRENT RESTART TIME STAMP OUT OF SEQUENCE - TIME=*date time* LOG RBA=*t-rba*

Severity
4

Explanation

This message indicates that the current log record has a time stamp that is less than the greatest time stamp processed so far. This might be a potential problem.

This message is followed by messages CSQ1147I and CSQ1148I which give the latest time stamp seen.

System action

Processing continues.

System programmer response

Examine the current log to determine whether multiple queue managers are writing to the same log. (Data might be being overwritten.) This might lead to data inconsistencies.

CSQ1146E

CURRENT END CHECKPOINT TIME STAMP OUT OF SEQUENCE - TIME=*date time* LOG RBA=*t-rba*

Severity
4

Explanation

This message indicates that the current log record has a time stamp that is less than the previous time stamp processed. This might be a potential problem.

This message is followed by messages CSQ1147I and CSQ1148I which give the latest time stamp seen.

System action

Processing continues.

System programmer response

Examine the current log to determine whether multiple queue managers are writing to the same log. (Data might be being overwritten.) This might lead to data inconsistencies.

CSQ1147I

LATEST TIME STAMP SEEN SO FAR - TIME=*date time* LOG RBA=*t-rba*

Severity
4

Explanation

This message follows message CSQ1145I or CSQ1146I and gives the latest time stamp seen.

CSQ1148I

MULTIPLE QUEUE MANAGERS MAY BE WRITING TO THE SAME LOG

Severity
4

Explanation

This message follows message CSQ1145I or CSQ1146I to indicate a possible cause of the time stamp problem.

CSQ1150I

SUMMARY OF COMPLETED EVENTS

Severity
0

Explanation

This message heads the summary of completed units of recovery (URs) and checkpoints.

System action

Processing continues.

CSQ1151I

UR CONNID=*cc* THREAD-XREF=*bb* USERID=*aa* TIME=*date time* START=*s-rba* END=*e-rba* DISP=*xx* INFO=*ii*

Severity
0

Explanation

This message describes a unit of recovery that terminated.

cc
Connection ID (for example, BATCH)

bb
Thread cross-reference ID (for example, JOB xxx)

aa
User ID executing the UR

date time
Starting time of the UR

s-rba
Log RBA of the first log record associated with the UR (that is, the URID)

e-rba
Log RBA of the last log record associated with the UR If the UR is not complete, *e-rba* is shown as '***'

xx
Disposition of the UR, values include:

- INFLIGHT
- IN BACKOUT
- IN COMMIT
- INDOUBT
- COMMITTED
- BACKED OUT

ii
Status of the data, one of the following:

- COMPLETE, indicating that all page sets modified by this UR have been identified
- PARTIAL, indicating that the list of page sets modified by this UR is incomplete (this is shown if all records associated with a UR are not available, and no checkpoint is found prior to the UR's completion)

If the UR identifying information is not available, it will be shown as '***'.

System action

Processing continues.

CSQ1153I

CHECKPOINT START=*s-rba* END=*e-rba* TIME=*date time*

Severity
0

Explanation

This message describes a complete checkpoint on the log starting at RBA *s-rba* and ending at RBA *e-rba*. If the information is available, CSQ1LOGP also returns the date and time that the checkpoint was completed.

When this message follows message CSQ1157I, it identifies the checkpoint that would be used at restart. If no checkpoint is available, message CSQ1158I is printed instead.

System action

Processing continues.

CSQ1154I

RESTART AT r-*rba* TIME=*date time*

Severity
0

Explanation

A normal restart occurred at log RBA *r-rba*. CSQ1LOGP also returns the date and time of that restart.

System action

Processing continues.

CSQ1155I

CONDITIONAL RESTART AT *r-rba* TIME=*date time*

Severity
0

Explanation

A conditional restart occurred at log RBA *r-rba*. CSQ1LOGP also returns the date and time of that restart.

System action

Processing continues.

CSQ1156I

ALL URS COMPLETE

Severity
0

Explanation

There are no URs outstanding for restart.

System action

Processing continues.

CSQ1157I

RESTART SUMMARY

Severity
0

Explanation

This message heads the summary of the description of work to be performed at restart. Restart information that follows is based on the scope of the log scanned. If you suspect an error in IBM MQ, see Problem determination on z/OS for information about identifying and reporting the problem.

System action

Processing continues.

CSQ1158I

NO CHECKPOINT AVAILABLE - RESTART SUMMARY INCOMPLETE

Severity
0

Explanation

No checkpoint is available within the scope of the log scanned. The information following this message includes:

- URs that have not completed
- Page sets modified by these URs
- Page sets with writes pending

The information cannot be considered complete.

System action

Processing continues.

CSQ1161E

INVALID URE FOUND AT *x-rba*

Severity
4

Explanation

While processing the SUMMARY option, an invalid URE checkpoint record was encountered in the log.

System action

Processing continues.

System programmer response

If the checkpoint record identified in the message is used to restart the queue manager, the restart will be unsuccessful because it will not be able to process the unit of recovery presented by the invalid URE.

Look for other messages that indicate the cause of the problem. If you are unable to resolve the problem, contact IBM Support.

CSQ1162E

INVALID RURE FOUND AT *x-rba*

Severity
4

Explanation

While processing the SUMMARY option, an invalid RURE checkpoint record was encountered in the log.

System action

Processing continues.

System programmer response

If the checkpoint record identified in the message is used to restart the queue manager, the restart will be unsuccessful because it will not be able to process the unit of recovery presented by the invalid RURE.

Look for other messages that indicate the cause of the problem. If you are unable to resolve the problem, contact IBM Support.

CSQ1163E

NO CHECKPOINT AVAILABLE DUE TO LOG ERROR - RESTART SUMMARY INCOMPLETE

Severity
4

Explanation

A log error was encountered. CSQ1LOGP marked any checkpoints encountered before the log error as invalid. There were no complete checkpoints following the log error in the specified log range. The information following this message includes:

- URs that have not completed
- Page set modified by these URs
- Page sets with writes pending

This information cannot be considered complete.

System action

Processing continues.

CSQ1165E

UR REQUIRES LOG WHICH IS IN ERROR

Severity
0

Explanation

While processing a UR, information was required from the log, but the log was in error, as indicated by previous messages.

System action

Processing continues.

CSQ1166I

INFORMATION INCOMPLETE FOR UR - LOG TRUNCATED AT *xx*

Severity
0

Explanation

Complete information for the UR is not available within the scope of the log scanned.

System action

Processing continues.

CSQ1209E

END OF LOG RANGE IS LESS THAN START

Severity
8

Explanation

The end log range value (specified by RBAEND or LRSNEND) is less than or equal to the start range value (specified by RBASTART or LRSNSTART).

System action

Processing is terminated.

System programmer response

Resubmit the job providing an RBASTART or LRSNSTART value and a corresponding RBAEND or LRSNEND value to specify a valid search range.

CSQ1210E

LOG READ ERROR RETCODE=*rc* REASON CODE=*reason*

Severity
8

Explanation

An error was detected while attempting to read the log.

System action

Processing is terminated.

CSQ1211E

BAD LOG RBA RETURNED

Severity
8

Explanation

One of the three problems listed in this topic exists:

- The recovery log data set is damaged
- You identified a data set that is not a recovery log data set
- There is a problem with the log print utility

System action

Processing terminates, and a dump is produced.

System programmer response

A common error is to specify the first data set on an archive tape (the Bxxxxxxx data set) as a log data set; it is actually a bootstrap data set (BSDS).

Determine if the problem is your error by dumping the data set and determining if it is a log data set.

CSQ1212I

FIRST LOG RBA ENCOUNTERED = *s-rba*

Severity
0

Explanation

This identifies the RBA of the first log record read.

System action

Processing continues.

CSQ1213I

LAST LOG RBA ENCOUNTERED = *e-rba*

Severity
0

Explanation

This identifies the RBA of the last log record read.

System action

Processing continues.

CSQ1214I

nn LOG RECORDS READ

Severity
0

Explanation

This identifies the number (in decimal) of logical log records read during CSQ1LOGP processing.

System action

Processing continues.

CSQ1215I

NO LOG RECORDS READ

Severity
0

Explanation

CSQ1LOGP read no log records.

Possible explanations are:

- An error has prevented CSQ1LOGP from continuing, therefore no log records have yet been processed (if this is so, an error message should precede this message)
- You specified the active log data sets or archive log data sets out of RBA sequence
- You specified an RBASTART or LRSNSTART value that is greater than any RBA or LRSN in the active and archive data sets available
- You specified a log range using LRSNs, but the queue manager is not in a queue sharing group.

System action

Processing continues.

CSQ1216E

LOG READ ERROR, RETCODE=*rc*, REASON CODE=*reason*, RBA=*x-rba*

Severity
4

Explanation

An error was encountered while attempting to read the log, indicating that either the log has an error in one of the control intervals (CI), or a data set containing the requested RBA cannot be located. The RBA specification in the message indicates where the error was detected and gives the requested RBA. It will point to:

- The start of the CI if there is a problem with the log control interval definition (LCID), or with any of the general control information within a CI
- The log record in the CI if there is a problem with a log record header (LRH)

If this is the first log record read during this execution of the Log Extractor, and if there is a problem with the LCID, the RBA specification will be all zeros.

Before returning any records, the utility checks the control information (LCID) at the end of a CI, and analyzes the LRH to ensure that all records are properly chained together within the CI. If an error is detected while performing this process, CSQ1LOGP will issue this message, before dumping the entire CI. It will not format individual records within the CI, but will, if possible, continue processing by reading the next CI.

System action

Processing continues.

CSQ1217E

RBA RANGE WARNING, RETCODE=*rc*, REASON CODE=*reason*, PRIOR RBA=*p-rba*, CURRENT RBA=*c-rba*

RBA RANGE WARNING, REASON CODE=*reason*, PRIOR RBA=*p-rba*, CURRENT RBA=*c-rba*

Severity
4

Explanation

A gap in the log RBA range has been encountered. PRIOR RBA *p-rba* indicates the last good log RBA prior to the gap. CURRENT RBA *c-rba* indicates the log record following the gap, and will be formatted following this message.

System action

Processing continues.

CSQ1218I

nn LOG ERROR MESSAGES

Severity
0

Explanation

CSQ1LOGP distinguishes three classes of errors:

- Code problems existing in the MQ or system code used for CSQ1LOGP. In such cases, abnormal termination with a user completion code of U0153 occurs.
- Incorrect invocation of CSQ1LOGP caused, perhaps, by your having used an incorrect keyword or missed a DD statement. Under these circumstances, CSQ1LOGP issues appropriate error messages, and the program is terminated.
- An error in a particular log CI under the scrutiny of CSQ1LOGP. Such scrutiny is performed before any of the records within the CI are processed. This is an indication of logical damage, and error messages are issued by the utility. The CI or log record in error is printed, and CSQ1LOGP continues to the next CI or log record.

  The count *nn* provided summarizes the number (in decimal) of errors CSQ1LOGP detected while accessing the log.

System action

Processing continues.

CSQ1219I

LOG RECORDS CONTAIN *n* BYTE RBA - QSG(*in-qsg*)

Severity
0

Explanation

This message is issued by CSQ1LOGP to indicate the format of the log records being processed, and whether the queue manager was a member of a queue sharing group (QSG). The message is issued before any log records are printed, and whenever the format of the log records change.

The value of *n* identifies the log RBA format of the log records being processed, and can be either 6 or 8.

The value of *in-qsg* identifies whether the log records were written by a queue manager that was a member of a QSG, and can be one of the following values:

YES
The log records were written by a queue manager that was a member of a QSG

NO
The log records were written by a queue manager that was not a member of a QSG

System action
Processing continues

CSQ1220E

ARCHIVE LOG TRUNCATED AT *xxxx* - INVALID LOG RECORDS READ

Severity
4

Explanation

At a restart of the queue manager, an archive log was truncated. This archive log data set could not be physically altered to reflect this truncation, and invalid log records therefore still exist. CSQ1LOGP has already reported this information in the summary report, and cannot retract it. Nor can it disregard the invalid log information already read in order adequately to summarize what has occurred. Therefore, all information up to this point in the log will be summarized, and a new summary report initiated. Consequently, the same UR might be reported twice with different dispositions and different page sets modified.

System action

Processing continues.

System programmer response

To avoid this condition, use the BSDS DD statement instead of the ARCHIVE DD statement.

CSQ1221E

VSAM ERROR, RETCODE=*rc*, REASON CODE=*reason*,VSAM RETURN CODE=*aaaa*, ERROR CODE=*bbbb*

Severity
8

Explanation

A VSAM error was encountered while attempting to read the log.

System action

Processing continues.

CSQ1222E

LOG ALLOCATION ERROR, RETCODE=*rc*, REASON CODE=*reason*, DYNALLOC INFO CODE=*aaaa*, ERROR CODE=*bbbb*

Severity
8

Explanation

An error occurred while dynamically allocating a log data set.

System action

Processing terminates.

CSQ1223E

JFCB READ ERROR, RETCODE=rc, REASON CODE=*reason*, RDJFCB RETURN CODE=*aaaa*

Severity
8

Explanation

An error occurred while trying to read the job file control block.

System action

Processing continues.

CSQ1224I

INFORMATION INCOMPLETE FOR LOG RECORD, CURRENT RBA=*c-rba*, CURRENT URID=*c-urid*

Severity
0

Explanation

Incomplete information for the log record was found within the scope of the logs scanned. An end of log condition was encountered before all segments of a spanned record could be found. CURRENT RBA *c-rba* indicates the log RBA of the record in question. CURRENT URID *c-urid* indicates the UR to which the spanned log record is related. If there is no URID associated with the log record (for instance, a checkpoint record), then this will show zeros.

System action

Processing continues.

System programmer response

If complete information for the identified log record is required, extend the RBA range to be processed until the required log data is available.

CSQ1271I

START OF LOG RANGE SET TO LRSN=*s-lrsn*

Severity
0

Explanation

The LRSN value you specified for the start of the log range is less than the lowest possible LRSN value, which is *s-lrsn*.

System action

Processing continues, using an LRSNSTART value of *s-lrsn*.

CSQ1272I

FIRST LOG LRSN ENCOUNTERED = *s-lrsn*

Severity
0

Explanation

This identifies the LRSN of the first log record read.

System action

Processing continues.

CSQ1273I

LAST LOG LRSN ENCOUNTERED = *e-lrsn*

Severity
0

Explanation

This identifies the LRSN of the last log record read.

System action

Processing continues.

CSQ1275I

LRSN RANGE CAN BE USED ONLY WITH A QUEUE SHARING GROUP

Severity
0

Explanation

You specified a log range using LRSNs, but CSQ1LOGP read no log records. This could be because the queue manager is not in a queue sharing group, in which case you cannot use LRSN specifications.

System action

Processing continues.

System programmer response

If the queue manager is not in a queue sharing group, rerun the job using RBA specifications for the log range.

CSQ1276E

LOG READ ERROR, RETCODE=*rc*, REASON CODE=*reason*, LRSN=*x-lrsn*

Severity
4

Explanation

An error was encountered while attempting to read the log, indicating that either the log has an error in one of the control intervals (CI), or a data set containing the requested LRSN cannot be located. The LRSN specification in the message indicates where the error was detected and gives the requested LRSN. It will point to:

- The start of the CI if there is a problem with the log control interval definition (LCID), or with any of the general control information within a CI
- The log record in the CI if there is a problem with a log record header (LRH)

If this is the first log record read during this execution of the Log Extractor, and if there is a problem with the LCID, the LRSN specification will be all zeros.

Before returning any records, the utility checks the control information (LCID) at the end of a CI, and analyzes the LRH to ensure that all records are properly chained together within the CI. If an error is detected while performing this process, CSQ1LOGP will issue this message, before dumping the entire CI. It will not format individual records within the CI, but will, if possible, continue processing by reading the next CI.

System action

Processing continues.

CSQ1277E

LRSN RANGE WARNING, RETCODE=*rc*, REASON CODE=*reason*, PRIOR LRSN=`p-lrsn`, CURRENT LRSN=*c-lrsn*

Severity
4

Explanation

A gap in the log LRSN range has been encountered. The PRIOR LRSN specification indicates the last good log LRSN prior to the gap. The CURRENT LRSN specification indicates the log record following the gap, and will be formatted following this message.

System action

Processing continues.

z/OS *IBM MQ-IMS bridge Messages (CSQ2...)*

CSQ2001I

csect-name OTMA REJECTED MESSAGE - APPLICATION ERROR, SENSE CODE=*code*, XCFGNAME=*gname* XCFMNAME=*mname* TPIPE=*tpipename*

Explanation

Because of an application error, the IBM MQ-IMS bridge received a negative acknowledgment (NAK) from OTMA when sending a message. The information provided in the message is:

gname
The XCF group to which the partner belongs.

mname
The member name of the partner.

tpipename
The name of the Tpipe used by the partner.

code
The IMS sense code returned by the partner (the first four characters are the sense code).

System action

The message is put to the dead-letter queue, and processing continues.

System programmer response

For information about IMS-OTMA sense codes, see the OTMA sense codes for NAK messages in the IMS product documentation.

CSQ2002E
csect-name OTMA CLIENT BID REJECTED, XCFGNAME=*gname* XCFMNAME=*mname*, SENSE CODE=*code*

Explanation

An OTMA client bid command from the IBM MQ-IMS bridge was rejected. *code* is the associated IMS sense code. *gname* and *mname* identify the partner IMS system to which the command was directed.

System action

No connection is made to the IMS system. Connections to other OTMA partners are unaffected.

System programmer response

For information about IMS-OTMA sense codes, see OTMA sense codes for NAK messages.

CSQ2003E
csect-name OTMA REJECTED MESSAGE - SYSTEM ERROR, SENSE CODE=*code*, XCFGNAME=*gname* XCFMNAME=*mname* TPIPE=*tpipename*

Explanation

Because of a system-related error, the IBM MQ-IMS bridge received a negative acknowledgment (NAK) from OTMA when sending a message. The information provided in the message is:

gname
The XCF group to which the partner belongs.

mname
The member name of the partner.

tpipename
The name of the Tpipe used by the partner.

code
The IMS sense code returned by the partner (the first four characters are the sense code).

System action

If the problem was caused by an environmental error, the IMS bridge returns the message to the queue. Depending on the error described by the sense code, the message send is retried or the queue is closed.

If a severe error occurred, the message is returned to the queue, and the IMS bridge ends abnormally with completion code X'5C6' and reason code X'00F20059'.

System programmer response

For information about IMS-OTMA sense codes, see OTMA sense codes for NAK messages.

CSQ2004E
csect-name ERROR USING QUEUE *q-name*, MQRC=*mqrc (mqrc-text)*

Explanation

The IBM MQ-IMS bridge was unable to open, close, get from, put to, or inquire about a queue.

If *csect-name* is CSQ2QCP0, the problem was with the message queue associated with IMS or the reply-to queue. If *csect-name* is CSQ2QCP1, the problem was with the reply-to queue. If *csect-name* is CSQ2PUTD, the problem was with the dead-letter queue.

If CSQ2PUTD fails to put a message in the dead-letter queue for some reason, the process which gets messages from the IMS bridge queue ends, and is only restarted by disabling and re-enabling the IMS Bridge queue, or by restarting the queue-manager.

If the reason code received is 2042, it is because the IBM MQ-IMS bridge requires exclusive input access (MQOO_INPUT_EXCLUSIVE) to the bridge queue if it is defined with QSGDISP(QMGR), or if it is defined with QSGDISP(SHARED) together with the NOSHARE option.

System action

If the problem was caused by an environmental error, processing continues.

If a severe error occurred, the IMS bridge ends abnormally with completion code X'5C6' and a reason code which shows the particular error.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about *mqrc* (*mqrc-text* provides the MQRC in textual form).

If *csect-name* is CSQ2PUTD, disable and re-enable the IMS bridge queue, and if that does not work, restart the queue-manager.

CSQ2005I
csect-name ERROR PROCESSING MESSAGE, FEEDBACK=*code*, XCFGNAME=*gname* XCFMNAME=*mname* TPIPE=*tpipename*

Explanation

The IBM MQ-IMS bridge encountered an error while processing a message. *code* is the associated feedback code that will be set in the message descriptor. The information provided in the message is:

gname
The XCF group to which the partner belongs.

mname
The member name of the partner.

tpipename
The name of the Tpipe used by the partner.

code
The IMS sense code returned by the partner.

If there is a loop of CSQ2005I messages causing FEEDBACK=292, see if there is a:

- Looping application that is putting badly-formed messages.
- Queue, for example the dead letter queue (DEADQ), that is not intended to be read by the IMS Bridge, but is using a STGCLASS with non-blank XCFGNAME and XCFMNAME parameters.

System action

The message is not processed.

System programmer response

code is one of the following:

291 (MQFB_DATA_LENGTH_ZERO)
A segment length field was zero in the application data of the message.

292 (MQFB_DATA_LENGTH_NEGATIVE)
A segment length field was negative in the application data of the message.

293 (MQFB_DATA_LENGTH_TOO_BIG)
A segment length field was too big in the application data of the message.

294 (MQFB_BUFFER_OVERFLOW)
The value of one of the length fields would overflow the MQ message buffer.

295 (MQFB_LENGTH_OFF_BY_ONE)
The length field was one byte too short.

296 (MQFB_IIH_ERROR)
The MQMD specified MQFMT_IMS, but the message does not begin with a valid MQIIH structure.

298 (MQFB_NOT_AUTHORIZED_FOR_IMS)
The user ID specified in the MQMD was denied access.

299 (MQFB_DATA_LENGTH_TOO_SHORT)
A segment length field was too short in the application data of the message.

3xx
IMS sense code xx (where xx is the decimal representation of the IMS sense code). For information about IMS-OTMA sense codes, see OTMA sense codes for NAK messages.

CSQ2006I
csect-name DEAD-LETTER QUEUE UNAVAILABLE, MQRC=*mqrc (mqrc-text)*

Explanation

The IBM MQ-IMS bridge was unable to put a message to the dead-letter queue.

System action

If the message was being sent to IMS, it will be retained on the local IMS queue, and the queue will be disabled. If the message was coming from IMS, a NAK will be sent to IMS so that IMS will retain it and stop sending messages on the Tpipe.

System programmer response

If *mqrc* is 0, there is no dead-letter queue defined; you are strongly recommended not to use the IBM MQ-IMS bridge unless you have a dead-letter queue defined. Otherwise, there is a problem obtaining the name of the queue from the queue manager; refer to "API completion and reason codes" on page 1151 for information about *mqrc* (*mqrc-text* provides the MQRC in textual form).

CSQ2007I
csect-name PROCESSING HAS STOPPED FOR IMS BRIDGE QUEUE *q-name*, XCFGNAME=*gname* XCFMNAME=*mname* TPIPE=*tpipename*

Explanation

This message follows message CSQ2004E, for *csect name* CSQ2PUTD failing to put a message in the dead-letter queue. The process which gets messages from the IMS bridge queue ends. The information provided in the message is:

q-name
The name of the IMS bridge queue.

gname
The XCF group to which the partner belongs.

mname
The member name of the partner

tpipename
The name of the Tpipe used by the partner

System action

Messages on this queue are no longer processed by the IMS bridge on this member.

System programmer response

To resume processing, GET-disable, and re-enable the IMS bridge queue or restart the Tpipe for the queue.

CSQ2009I
csect-name PREREQUISITE PRODUCTS FOR IMS BRIDGE NOT AVAILABLE

Explanation

The IBM MQ-IMS bridge cannot operate because:

- The version of z/OS being used is not correct
- The version of IMS being used is not correct
- OTMA support has not been enabled on IMS.
- An incorrect version of the system parameter module (CSQZPARM) is being used.

System action

The MQ-IMS bridge does not start.

System programmer response

Refer to Planning your IBM MQ environment on z/OS for information about what product levels are required.

If required, recompile CSQZPARM with the correct libraries.

CSQ2010I
csect-name CONNECTED TO PARTNER, XCFGNAME=*gname* XCFMNAME=*mname*

Explanation

The MQ-IMS bridge successfully established a connection to the partner IMS system identified by *gname* and *mname*.

System action

Processing continues; messages can be sent to the partner.

CSQ2011I
csect-name DISCONNECTED FROM PARTNER, XCFGNAME=*gname* XCFMNAME=*mname*

Explanation

The partner IMS system identified by *gname* and *mname* is no longer available, and the connection from the IBM MQ-IMS bridge has ended.

System action

Processing continues; messages can no longer be sent to the partner.

CSQ2012I
csect-name NO UTOKEN SECURITY REQUESTED FOR IMS SIGNON, XCFGNAME=*gname* XCFMNAME=*mname*

Explanation

The IBM MQ-IMS bridge signed-on to the partner IMS system identified by *gname* and *mname*. No UTOKEN security was requested for this session.

System action

Processing continues.

CSQ2013E
csect-name NOT AUTHORIZED FOR IMS SIGNON, XCFGNAME=*gname* XCFMNAME=*mname*

Explanation

The IBM MQ-IMS bridge tried to sign on to the partner IMS system identified by *gname* and *mname*. However, the queue manager not authorized to establish a connection to this IMS system.

System action

No connection is made to the IMS system. Connections to other OTMA partners are unaffected.

CSQ2015I
csect-name IMS BRIDGE ALREADY SUSPENDED, XCFGNAME=*gname* XCFMNAME=*mname*

Explanation

A SUSPEND QMGR FACILITY(IMSBRIDGE) command was issued, but the IBM MQ-IMS bridge to the partner IMS system identified by *gname* and *mname* is already suspended.

System action

None.

CSQ2016I
csect-name IMS BRIDGE NOT SUSPENDED, XCFGNAME=*gname* XCFMNAME=*mname*

Explanation

A RESUME QMGR FACILITY(IMSBRIDGE) command was issued, but the IBM MQ-IMS bridge to the partner IMS system identified by *gname* and *mname* is not suspended.

System action

None.

CSQ2020E
csect-name RESYNCHRONIZATION ERROR

Explanation

A resynchronization error has occurred. The information provided by this message is:

```
  IN TPIPE tpipename  FOR QUEUE q-name,  BY PARTNER, XCFGNAME=gname XCFMNAME=mname,
QMGR SEND=sendseq PARTNER RECEIVE=otmarecvseq,QMGR RECEIVE=recvseq
 PARTNER SEND=otmasendseq,  INDOUBT UNIT OF RECOVERY urid
```

where:

tpipename
The name of the Tpipe which cannot be resynchronized

q-name
The name of the queue for this Tpipe

gname
The name of the XCF group to which the Tpipe belongs

mname
The name of the XCF member to which the Tpipe belongs

sendseq
The recoverable sequence number of the message last sent by IBM MQ to the partner, in hexadecimal

otmasendseq
The recoverable sequence number of the message last sent by the partner to IBM MQ, in hexadecimal

recvseq
The recoverable sequence number of the message last received by IBM MQ from the partner, in hexadecimal

otmarecvseq
The recoverable sequence number of the message last received by the partner from IBM MQ, in hexadecimal

urid
The identifier of an in-doubt unit of recovery; a value of 0 means that there is no in-doubt unit of recovery.

System action

No messages are sent on the Tpipe.

System programmer response

Use the RESET TPIPE command to reset recoverable sequence numbers, to restart the Tpipe, and, if required, to resolve the unit of recovery.

CSQ2023E
csect-name PARTNER, XCFGNAME=*gname* XCFMNAME=*mname*, CANNOT RESYNCHRONIZE, SENSE CODE=*code*

Explanation

IBM MQ was unable to resynchronize with the partner. The information provided in the message is:

gname
The name of the XCF group to which the partner belongs.

mname
The member name of the partner who cannot resynchronize.

code
The IMS sense code returned by the partner (the first four characters are the sense code).

System action

The connection to OTMA is stopped

System programmer response

For information about IMS-OTMA sense codes, see OTMA sense codes for NAK messages. Resolve the problem and restart the OTMA connection.

CSQ2024E
csect-name TPIPE *tpipename* IS UNKNOWN TO PARTNER, XCFGNAME=*gname* XCFMNAME=*mname*

Explanation

The Tpipe name was unknown to the partner. The information provided in the message is:

tpipename
The name of the Tpipe which the partner no longer recognizes.

gname
The XCF group to which the partner belongs.

mname
The member name of the partner who is resynchronizing

System action

The associated unit of recovery is backed out and processing continues.

System programmer response

If the partner IMS system has been cold started then this message can be considered normal. If the IMS system has not been cold started consider this message as an alert and investigate the partner IMS system.

CSQ2025E
csect-name PARTNER, XCFGNAME=*gname* XCFMNAME=*mname*, CANNOT RESYNCHRONIZE TPIPE *tpipename*, SENSE CODE=*code*

Explanation

The partner was unable to resynchronize the Tpipe. The information provided in the message is:

gname
The XCF group to which the partner belongs.

mname
The member name of the partner who is resynchronizing.

tpipename
The name of the Tpipe which the partner cannot resynchronize.

code
The IMS sense code returned by the partner.

System action

The Tpipe is stopped.

System programmer response

See OTMA sense codes for NAK messages for information about the sense code from IMS. Resolve the problem and restart or reset the Tpipe.

CSQ2026I
csect-name PARTNER, XCFGNAME=*gname* XCFMNAME=*mname*, HAS COLD-STARTED TPIPE *tpipename*

Explanation

The partner has cold started a Tpipe. The information provided in the message is:

gname
The XCF group of which the partner is a member.

mname
The member name of the partner who is resynchronizing.

tpipename
The name of the Tpipe which the partner has cold started.

System action

All recoverable sequence numbers are reset to 1, and processing continues.

System programmer response

None.

CSQ2027I
csect-name TPIPE *tpipename* FOR PARTNER, XCFGNAME=*gname* XCFMNAME=*mname*, DOES NOT HAVE AN INDOUBT UNIT OF RECOVERY

Explanation

MQ expected a Tpipe to have an in-doubt unit of recovery. The information provided by the message is:

tpipename
The name of the Tpipe for which there should be a unit of recovery still in doubt

gname
The XCF group to which the partner belongs.

mname
The member name of the partner for the Tpipe.

System action

Processing continues.

System programmer response

Collect the following items, and contact IBM Support.

- Console log
- MQ job log
- IMS job log

CSQ2028I
csect-name QUEUE MANAGER IS NOT CONNECTED TO PARTNER, XCFGNAME=*gname* XCFMNAME=*mname*

Explanation

MQ is not connected to the partner. The information provided in the message is:

gname
The group name of the partner.

mname
The member name of the partner.

System action

The command is rejected.

System programmer response

Resubmit the command using the correct XCF group name when IBM MQ is connected to the partner.

CSQ2029I
csect-name TPIPE *tpipename* NOT FOUND FOR PARTNER, XCFGNAME=*gname* XCFMNAME=*mname*

Explanation

The Tpipe could not be found. The information provided in this message is:

tpipename
The name of the Tpipe which could not be found.

gname
The XCF group of which the partner is a member.

mname
The member name of the partner for the Tpipe.

System action

The command is rejected.

System programmer response

Resubmit the RESET TPIPE command with the correct Tpipe name.

CSQ2030I
csect-name TPIPE *tpipename* IS STILL OPEN FOR PARTNER, XCFGNAME=*gname* XCFMNAME=*mname*

Explanation

The Tpipe is still open. The information provided by this message is:

tpipename
The name of the Tpipe which is still open.

gname
The XCF group name.

mname
The member name of the partner for the Tpipe.

System action

The command is rejected.

System programmer response

The most likely cause of this message is that the RESET TPIPE command was issued with an incorrect Tpipe name or that the command was issued on the wrong queue manager in a queue sharing group. Resubmit the RESET TPIPE command with the correct Tpipe name.

CSQ2031I

csect-name TPIPE *tpipename* FOR PARTNER, XCFGNAME=*gname* XCFMNAME=*mname*, ACTION REQUIRED FOR INDOUBT UNIT OF RECOVERY

Explanation

A Tpipe has an in-doubt unit of recovery, but no recovery action was specified. The information provided by the message is:

tpipename
The name of the Tpipe which has a unit of recovery still in doubt

gname
The XCF group to which the partner belongs.

mname
The member name of the partner for the Tpipe.

System action

Processing continues.

System programmer response

Resubmit the RESET TPIPE command specifying an action (COMMIT or BACKOUT) for the in-doubt unit of recovery.

CSQ2040I

csect-name OTMA MESSAGE FLOOD STATUS=WARNING FOR PARTNER, XCFGNAME=*gname* XCFMNAME=*mname*

Severity
4

Explanation

This message is issued by the IBM MQ-IMS bridge in response to a notification from the partner IMS system, identified by *gname* and *mname*, that an OTMA message flood warning condition exists.

This message indicates that the IMS partner is currently unable to process the volume of transaction requests being sent to it via the IBM MQ-IMS bridge.

System action

Processing continues but the IBM MQ-IMS bridge will slow down the rate at which transaction requests are sent to allow the partner IMS system to process the accumulated backlog.

System programmer response

Review the status of the partner IMS system to determine if any action is required. You can use the **/DISPLAY OTMA** and **/DISPLAY TMEMBER** commands to do this.

Perform a check on the partner IMS system to determine if the message DFS1988W has been issued, identifying the severity of the warning condition.

CSQ2041I

csect-name OTMA MESSAGE FLOOD STATUS=FLOODED FOR PARTNER, XCFGNAME=*gname* XCFMNAME=*mname*

Severity
8

Explanation

This message is issued by the IBM MQ-IMS bridge in response to a notification from the partner IMS system, identified by *gname* and *mname*, that an OTMA message flood condition exists.

This indicates that the IMS partner is currently unable to process the volume of transaction requests being sent to it through the IBM MQ-IMS bridge. No further requests can be sent until the flood condition in IMS has been relieved.

System action

All TPIPEs to the identified partner IMS system are suspended until a notification is received from IMS indicating that the flood condition has been relieved.

Messages can still be put to any IBM MQ-IMS bridge queue with a storage class specifying the identified IMS partner but will remain there until the TPIPES can be resumed.

IBM MQ-IMS bridge queues for other IMS partners are unaffected.

System programmer response

Review the status of the partner IMS system and determine what action is required to relieve the IMS flood condition. You can use the **/DISPLAY OTMA** and **/DISPLAY TMEMBER** commands to do this.

Perform a check on the partner IMS system to determine if the message DFS1989E has been issued, identifying the flood condition.

CSQ2042I

csect-name OTMA MESSAGE FLOOD RELIEVED FOR PARTNER, XCFGNAME=*gname* XCFMNAME=*mname*

Severity

0

Explanation

This message is issued by the IBM MQ-IMS bridge in response to a notification from the partner IMS system, identified by *gname* and *mname*, that an OTMA message flood, or flood warning, condition no longer exists.

System action

If this message follows CSQ2041I, all TPIPEs to the identified partner IMS system that were suspended in response to the flood condition are resumed. The IBM MQ-IMS bridge will gradually increase the rate at which transaction requests are sent until the maximum rate is achieved, or a subsequent flood condition is reported by the partner IMS system.

System programmer response

None required.

z/OS *Subsystem support messages (CSQ3...)*

CSQ3001E

csect-name - ABNORMAL DISCONNECT FROM SUBSYSTEM INTERFACE

Explanation

An online routine was still supporting SSI calls (IEFSSREQ) even though the queue manager had nearly completed termination or was no longer executing. This occurs with *csect-name* CSQ3RS00 or CSQ3RS0X when the queue manager address space has reached end-of-memory and neither normal termination nor online error recovery routines have successfully completed termination of the queue manager. This occurs with *csect-name* CSQ3SSTM when this condition is discovered during online termination.

System action

The connection is terminated. All IEFSSREQ requests are handled by the IBM MQ early processing program until the queue manager is restarted. An SVC dump is requested.

CSQ3002I
INDOUBT RECOVERY BY *connection-name* STILL IN PROGRESS

Explanation

There might be IBM MQ units of recovery (URs), related to an identified subsystem (*connection-name*), still in doubt after restart synchronization has taken place. (Indoubt URs are those for which commit has been voted by IBM MQ but which have not yet been acknowledged by *connection-name*.)

This message might appear if the *connection-name* subsystem has begun to do new work before having resolved all in-doubt URs. The *connection-name* subsystem is still in the process of resolving the in-doubt URs.

System action

Resources held (locked) by these in-doubt URs are unavailable to any other work units until their status is resolved.

System programmer response

The system programmer or system administrator must determine the correct recovery action to resolve the in-doubt situations. This involves either ensure-commit or backout decisions for all in-doubt URs.

The DISPLAY THREAD command should be used to see the URs still in doubt. It will normally show that all in-doubt URs have now been resolved. If not, the RESOLVE INDOUBT command should be used to resolve the in-doubt URs and to release the resources they hold.

CSQ3004E
SSI DESCRIPTOR GET FAILURE, RC=*rc* REASON=*reason*

Explanation

An internal error has occurred during initialization or termination.

System action

The queue manager terminates.

System programmer response

Ensure that all maintenance has been applied to the IBM MQ program libraries, and then restart the queue manager.

CSQ3006E
'*rmid*' SSI FUNCTION WAS ALREADY ACTIVE WHEN ACTIVATE WAS ATTEMPTED

Explanation

An initialization sequence error has occurred.

System action

The queue manager terminates.

System programmer response

Ensure that all maintenance has been applied to the IBM MQ program libraries, and then restart the queue manager.

CSQ3007E
'*rmid*' SSI FUNCTION WAS ALREADY INACTIVE WHEN DEACTIVATE WAS ATTEMPTED

Explanation

A termination sequence error has occurred.

System action

Termination continues.

System programmer response

Ensure that all maintenance has been applied to the IBM MQ program libraries.

CSQ3008E
csect-name - ABNORMAL DISCONNECT FOR PROGRAM REQUEST HANDLER(S)

Explanation

One or more resource managers are still supporting application program calls through their program request handler, even though the queue manager had almost completed termination, or was no longer executing. This occurs when the queue manager address space has gone to end of memory and neither normal termination nor online error recovery routines have successfully completed termination.

System action

The connection is terminated. All application program support requests are rejected with an indication that the queue manager is not active. An SVC dump is requested.

System programmer response

If the problem persists, collect the following items, and contact IBM Support:

- System dump
- Printout of SYS1.LOGREC

CSQ3009E
error-info

Explanation

An internal error has occurred in RRS exit processing. The message contains error information that will be needed to resolve the problem.

System action

Processing continues, but RRS coordination is no longer available to the queue manager. It will probably be necessary to restart the queue manager or RRS.

CSQ3011I
csect-name Coordinator RRS is cold-starting and has lost its log. In-doubt IBM MQ threads need manual resolution

Explanation

IBM MQ has participant responsibility for in-doubt threads. RRS, the commit coordinator, has informed the queue manager that it lost all knowledge of IBM MQ in-doubt threads. The in-doubt threads at this queue manager must be manually resolved with the RESOLVE INDOUBT command.

System action

Processing continues.

System programmer response

A list of in-doubt threads where RRS is the coordinator can be displayed using the DISPLAY THREAD command for in-doubt type threads by specifying RRSBATCH as the connection name.

The decision to commit or back out the logical unit of work should be coordinated with any other participant RRS Recoverable Resource Managers. The existence of other participants might not be easy to determine. The information might be available in the RRS recovery log even though information has been lost.

At this queue manager, all in-doubt threads coordinated by RRS must be resolved with the RESOLVE INDOUBT command. Locked data remains unavailable until resolution. Threads that were already

resolved with this command are discarded. Threads not yet resolved are discarded after resolution with the command.

The commit or back out decision provided using the RESOLVE INDOUBT command for a logical unit of work is propagated to all downstream participants, if any.

CSQ3013I
csect-name Queue manager was restarted on the wrong system so cannot connect to RRS. There are unresolved URs where IBM MQ is a participant

Explanation

The queue manager has one or more in-doubt threads and is unable to connect to RRS to resolve these in-doubt units of recovery (URs).

System action

Processing continues.

CSQ3014I
csect-name In-doubt RRS URID=rrs-*urid* is unknown to IBM MQ. URID recorded for IBM MQ by RRS=mq-*urid*

Explanation

The queue manager is restarting with RRS where IBM MQ is a participant and RRS is the coordinator. RRS has a unit of recovery (UR) that the queue manager should be a participant in, but it has no knowledge of the RRS unit of recovery, with an ID of *rrs-urid*. RRS has recorded the IBM MQ URID as *mq-urid*.

System action

Restart with RRS continues.

System programmer response

This message might indicate a problem in IBM MQ or RRS, or it might be produced because of one of the following prior actions:

- A conditional restart was performed that resulted in the loss of part or all of the IBM MQ log. This conditional restart might have happened at any time in the past.
- The RESOLVE INDOUBT command was used to resolve the IBM MQ UR with ID *mq-urid*.

If one of these occurred, the message can be ignored. If neither occurred, there might be a problem in IBM MQ or RRS.

If the *mq-urid* appears to be a valid log RBA, use the log print utility (CSQ1LOGP) with the SUMMARY option and URID options using the *mq-urid* value. If this finds the UR, the disposition will indicate whether it was committed or backed out. If possible, use the RRS ISPF interface to commit or back out the RRS URID so that they match.

If you suspect an error in IBM MQ, collect the items listed in the Problem Determination section and contact IBM Support.

CSQ3016I
csect-name RRS has lost data from its log

Explanation

The queue manager is restarting with RRS and RRS has lost some portion of its log.

System action

Restart with RRS continues.

System programmer response

IBM MQ might not be able to resolve in-doubt units of recovery successfully with RRS because of the loss of RRS log data.

CSQ3017I
csect-name RRS function *call-name* failed, RC=*rc*

Explanation

During queue manager restart, the RRS function specified by *call-name* issued a return code *rc* indicating a failure.

System action

Processing continues, but RRS functions will not be available. For example, connections using the RRS adapter will not be allowed, and queue sharing group facilities will not work.

System programmer response

Investigate the RRS return code from the function specified and resolve the problem. Then restart the queue manager.

CSQ3018I
csect-name RRS function synchronization complete

Explanation

The queue manager has completed synchronization processing with RRS, and RRS functions are available.

System action

None.

System programmer response

None.

CSQ3100I
csect-name - SUBSYSTEM *ssnm* READY FOR START COMMAND

Explanation

The queue manager has terminated, and can be restarted when required.

CSQ3101E
csect-name - INVALID EARLY PROCESSING PARAMETER

Explanation

The z/OS command SETSSI ADD or the subsystem definition record in the IEFSSNxx member of SYS1.PARMLIB for the IBM MQ subsystem specified the early processing initialization parameter incorrectly. The name must be CSQ3EPX.

The failing subsystem name is provided in message IEF759I, which follows this message.

System action

The IBM MQ subsystem with the indicated name is not available.

System programmer response

Correct the parameter fields in the record of SYS1.PARMLIB member IEFSSNxx. For information about the parameters, see Update SYS1.PARMLIB members.

CSQ3102E
csect-name - INVALID COMMAND PREFIX

Explanation

The z/OS command SETSSI ADD or the subsystem definition record in the IEFSSNxx member of SYS1.PARMLIB for the IBM MQ subsystem specified the command prefix initialization parameter incorrectly.

The failing subsystem name is provided in message IEF759I, which follows this message.

System action

The IBM MQ subsystem with the indicated name is not available.

System programmer response

Correct the parameter fields in the record of SYS1.PARMLIB member IEFSSNxx. For information about the parameters, see Update SYS1.PARMLIB members.

CSQ3104I
csect-name - TERMINATION COMPLETE

Explanation

The queue manager has terminated. The actual z/OS termination of the queue manager address spaces might have completed earlier. This message is presented for every termination, normal or abnormal.

CSQ3105E
csect-name - UNABLE TO LOAD EARLY PROCESSING PROGRAM 'CSQ3EPX'. *ssnm* IS NOT AVAILABLE

Explanation

Subsystem initialization or early processing refreshing for the IBM MQ subsystem failed because the initialization program (CSQ3INI) could not locate the early processing program (CSQ3EPX).

For subsystem initialization, the program must be either in the linkpack area (LPA) or in a library which is in the link list. For early processing refreshing, the program must be in the LPA.

System action

Subsystem initialization or early processing refreshing ends abnormally. IBM MQ subsystem *ssnm* is not available.

CSQ3106E
csect-name - QUEUE MANAGER STOPPED. COMMAND NOT PROCESSED - *command-text*

Explanation

A command was received which cannot be processed due to one of the following:

- The queue manager has not been started (this could be because the START QMGR command was not entered correctly)
- The command was queued for processing while the queue manager was starting, but startup terminated with an error
- The queue manager terminated before the command could be processed

System action

The command is not processed.

CSQ3107E
csect-name - COMMAND REJECTED. REQUESTER NOT AUTHORIZED

Explanation

A command was received from a console that does not have the correct authority.

System action

The command is not processed. This message is sent to the console that entered the command.

System programmer response

Verify that this console should be used for entering IBM MQ commands. If so, authorize it for IBM MQ commands by using z/OS services.

Note: If IBM MQ security is not activated, this check is still performed. This authorization is the z/OS console authority, and is not related to the external security manager. The user ID that entered the IBM MQ command must have OPERPARM AUTH with SYS, ALL, or MASTER console authority.

CSQ3108E
csect-name - COMMAND REJECTED. COMMAND FACILITY PATH UNAVAILABLE

Explanation

A command was received, but the path from z/OS consoles to the IBM MQ command processor is unavailable. It might still be possible to enter commands in other ways. You can also receive this message if the early code for the queue manager was being refreshed when the command was issued.

System action

The command is not processed. This message is delivered to the console that entered the command.

System programmer response

The console command facility is available again the next time the queue manager is started.

If the command was rejected because the early code for the queue manager was being refreshed when you issued it, wait until message CSQ3110I is issued to indicate that the early code has successfully refreshed before you issue the command again.

CSQ3109E
csect-name - UNABLE TO OBTAIN SUBSYSTEM AFFINITY TABLE INDEX FOR SUBSYSTEM *ssnm*. IEFSSREQ RC=*nn*

Explanation

IBM MQ was unable to obtain a subsystem affinity table index for the named subsystem. z/OS did not recognize the named subsystem name as a known subsystem. If this message is issued, a serious error has occurred in z/OS or IBM MQ.

In the message, *nn* is the return code from the IEFSSREQ z/OS service. *ssnm* is the name of the IBM MQ subsystem undergoing IPL-time initialization.

System action

IBM MQ ends abnormally with completion code X'5C6' and reason code X'00F30104'. The IBM MQ subsystem with the indicated name is not available for this IPL of z/OS.

System programmer response

Try to perform an IPL of the z/OS system. If the problem persists, see Problem determination on z/OS for information about identifying and reporting the problem.

CSQ3110I
csect-name - SUBSYSTEM *ssnm* INITIALIZATION COMPLETE

Explanation

Either:

- IBM MQ subsystem initialization is complete, following z/OS IPL processing or the z/OS command SETSSI ADD.
- The IBM MQ early processing program has been successfully refreshed, following a REFRESH QMGR TYPE(EARLY) command.

CSQ3111I
csect-name - EARLY PROCESSING PROGRAM IS V*n* LEVEL *l*

Explanation

This message shows the level of the early processing program that is being used.

The level is of the form *nnn-mmm* and indicates the capability of the early code.

nnn can be incremented between releases (but not every release increments it) and *mmm* can be incremented from time to time when PTFs add maintenance to the early code. Increments to *mmm* only relate to the IBM MQ version reported. For example "V9.2 level 010-008" and "V9.3 level 010-008" are not equivalent.

The early code level used must have a capability level corresponding with the highest release of the product you intend to run on an LPAR. You can use the *nnn* value to confirm the level installed.

Corresponding values of *nnn* are:

- **006**: IBM WebSphere MQ for z/OS 7.1
- **007**: IBM MQ for z/OS 8.0
- **008**: IBM MQ for z/OS 9.0
- **009**: IBM MQ for z/OS 9.1
- **010**: IBM MQ for z/OS 9.2
- **010**: IBM MQ for z/OS 9.3
- **010**: IBM MQ for z/OS 9.4

CSQ3112E
csect-name - INVALID CPF SCOPE

Explanation

The z/OS command SETSSI ADD or the subsystem definition record in the IEFSSNxx member of SYS1.PARMLIB for the IBM MQ subsystem specified the CPF scope initialization parameter incorrectly.

The failing subsystem name is provided in message IEF759I, which follows this message.

System action

The IBM MQ subsystem with the indicated name is not available.

System programmer response

Correct the parameter fields in the record of SYS1.PARMLIB member IEFSSNxx. For information about the parameters, see Update SYS1.PARMLIB members.

CSQ3113E
csect-name - COMMAND PREFIX REGISTRATION FAILED. INVALID CHARACTER(S) IN CPF

Explanation

Command prefix registration failed because the command prefix (CPF) contains invalid characters.

System action

The IBM MQ subsystem with the indicated name is not available.

System programmer response

Correct the CPF parameter in the record of SYS1.PARMLIB member IEFSSNxx. For information about the parameters, see Update SYS1.PARMLIB members.

CSQ3114E
csect-name - COMMAND PREFIX REGISTRATION FAILED. INVALID CHARACTER(S) IN SUBSYSTEM NAME

Explanation

Command prefix registration failed because the subsystem name used as the owner of the command prefix (CPF) contains invalid characters.

System action

The IBM MQ subsystem with the indicated name is not available.

System programmer response

Correct the CPF parameter in the record of SYS1.PARMLIB member IEFSSNxx. For information about the parameters, see Update SYS1.PARMLIB members.

CSQ3115E
csect-name - COMMAND PREFIX REGISTRATION FAILED. CPF ALREADY DEFINED

Explanation

Command prefix registration failed because the command prefix (CPF) was already defined to z/OS.

System action

The IBM MQ subsystem with the indicated name is not available.

System programmer response

Correct the CPF parameter in the record of SYS1.PARMLIB member IEFSSNxx. For information about the parameters, see Update SYS1.PARMLIB members.

CSQ3116E
csect-name - COMMAND PREFIX REGISTRATION FAILED. CPF IS A SUBSET OF A CPF ALREADY DEFINED

Explanation

Command prefix registration failed because the command prefix (CPF) is a subset of a CPF already defined to z/OS.

System action

The IBM MQ subsystem with the indicated name is not available.

System programmer response

Correct the CPF parameter in the record of SYS1.PARMLIB member IEFSSNxx. For information about the parameters, see Update SYS1.PARMLIB members.

CSQ3117E
csect-name - COMMAND PREFIX REGISTRATION FAILED. CPF IS A SUPERSET OF A CPF ALREADY DEFINED

Explanation

Command prefix registration failed because the command prefix (CPF) is a superset of a CPF already defined to z/OS.

System action

The IBM MQ subsystem with the indicated name is not available.

System programmer response

Correct the CPF parameter in the record of SYS1.PARMLIB member IEFSSNxx. For information about the parameters, see Update SYS1.PARMLIB members.

CSQ3118E
csect-name - SYSTEM ERROR DURING COMMAND PREFIX REGISTRATION

Explanation

A z/OS error occurred during command prefix (CPF) registration.

System action

The MQ subsystem with the indicated name is not available.

System programmer response

Check the z/OS console for other messages relating to the problem.

CSQ3119E
csect-name call-name call for group attach table failed, rc=*rc*

Explanation

During initialization for the group connect facility, a name token services call failed. *rc* is the return code (in hexadecimal) from the call.

System action

Processing continues, but the group connect facility will not be available to CICS.

System programmer response

See the *z/OS MVS Authorized Assembler Services Reference* manual for information about the return codes from the IEANTCR name token services call. If you are unable to solve the problem, take a stand-alone system dump and contact IBM Support.

CSQ3120E
csect-name - IXCQUERY ERROR FOR XCF GROUP *group-name* APPLID= *applid*, RC= *rc* REASON= *reason*

Explanation

A CICS region with APPLID *applid* attempted to connect to a queue sharing group. During processing of the request an IXCQUERY call failed with return code *rc* and reason code *reason*.

The XCF group for which the IXCQUERY request was performed is identified by *group-name*.

System action

The request by CICS to connect to the queue sharing group fails with the reason code MQRC_UNEXPECTED_ERROR.

System programmer response

See the *z/OS MVS Sysplex Services Reference* manual for an explanation of the IXCQUERY return and reason codes. If you are unable to solve the problem, contact IBM Support.

CSQ3201E
ABNORMAL EOT IN PROGRESS FOR USER=*user* CONNECTION-ID=*conn-id* THREAD-XREF=*thread-xref* JOBNAME=*jobname* ASID=*asid* TCB=*tcb*

Explanation

Abnormal termination processing has been started for the agent with the values for the USER, CONNECTION-ID, THREAD-XREF, JOBNAME, ASID and TCB shown. These values are the last known set of identifiers for the terminating agent.

The abnormal termination might be the result of an error in the allied agent's address space or the result of the z/OS command CANCEL issued by the operator.

The value for the USER, the THREAD-XREF or both might be blank. The values for the USER, CONNECTION-ID, THREAD-XREF, JOBNAME and ASID are the last values established to IBM MQ for this connection and might represent the current activity of the agent. The TCB value is the address of the TCB that is terminating. Previous IBM MQ work by this agent might have completed successfully.

This message, CSQ3201E, is written to the z/OS console after the agent has been removed from the service task work queue at the time that termination processing begins.

System action

The agent was previously queued to a service task for termination processing. This message indicates that the agent has been taken from the queue for processing. Any uncommitted changes will be backed out.

System programmer response

See the Problem Determination section of this message. The z/OS commands CANCEL and FORCE will have no effect. Do not cancel IBM MQ. If an extensive backout is in progress, the subsequent queue manager restart might take a very long time due to additional log activity.

CSQ3202E
CONNECTION FOR *jobname* FAILED, INSUFFICIENT ECSA STORAGE TO CREATE ACE

Explanation

jobname attempted to connect to IBM MQ using the MQCONN, or MQCONNX, API call.

There was insufficient common storage available to build the control blocks to represent the connection and to the connection attempt failed.

There might be a system wide ECSA shortage, or the storage available for creating new queue manager connections might be limited by the ACELIM system parameter.

This message can be seen for batch applications, including RRS applications; for example, Db2 stored procedures and WebSphere Application Server.

System action

The MQCONN or MQCONNX API call, used by *jobname* returns MQCC_FAILED, together with reason code MQRC_Q_MGR_NOT_AVAILABLE 2059

Queue manager processing continues.

CSQ3580E
CONNECTION FOR '*ssi-call*' GAVE RC=*rc*, REASON=*reason*

Explanation

A nonzero return code has been returned to CSQ3AMI2 from the connect to subsystem interface (SSI) call. The variables in the message indicate which SSI call is involved and the actual return and reason codes associated with it.

System action

The current task is ended abnormally with a system completion code of X'5C6' and a reason code of X'00F30580'. The queue manager terminates.

System programmer response

Restart the queue manager. Note the values contained in the message, and contact IBM Support.

z/OS *Db2 manager messages (CSQ5...)*

CSQ5001I
csect-name Connected to Db2 *db2-name*

Explanation

The queue manager has successfully established a connection to the named Db2 subsystem.

System action

Processing continues.

System programmer response

None.

CSQ5002E
csect-name Connection to Db2 using *connect-name* failed, RC=*return-code* reason=*reason*

Explanation

The queue manager's attempt to establish a connection to the named Db2 subsystem failed.

System action

Queue manager startup is terminated.

System programmer response

This is normally an authorization error.

See Db2 codes in the *Db2 for z/OS* documentation for an explanation of the codes and attempt to resolve the problem.

CSQ5003A
csect-name Connection to Db2 using *connect-name* pending, no active Db2

Explanation

The queue manager is waiting for an eligible Db2 subsystem to become active so that a connection can be established. Alternatively, RRS is inactive or was started after the Db2 subsystems.

System action

The queue manager waits for an eligible Db2 subsystem to become active.

System programmer response

Check whether the Db2 subsystem(s) are active. If not then start them. If they are active, ensure RRS is active and check that it was started prior to the Db2 subsystems.

CSQ5004E
csect-name Db2 table entry for queue manager in queue sharing group *qsg-name* is missing or incorrect

Explanation

During startup the queue manager was unable to find its entry in the Db2 administration tables, or the entry was incorrect.

System action

The queue manager terminates with completion code X'6C6' and reason code X'00F50013'.

System programmer response

Check that a queue manager record exists in the Db2 tables for the Db2 data-sharing group specified. Check the QSGDATA system parameter specifies the correct Db2 data-sharing group. If so, check that a queue manager entry exists in the CSQ.ADMIN_B_QMGR table.

If you are migrating from a previous release of IBM MQ, check also that you have updated the Db2 tables to the format for the current release. For information about migration and compatibility between releases, see Maintaining and migrating.

CSQ5005E
csect-name Queue manager release level is incompatible with queue sharing group

Explanation

The release level of the queue manager that is being started is incompatible with that of other members of the queue sharing group.

System action

The queue manager terminates with completion code X'6C6' and reason code X'00F50029'.

System programmer response

For information about migration and compatibility between releases, see Queue sharing group migration.

Locate the queue managers in the queue-sharing group that are below the level supported by the queue manager issuing the message, and upgrade them to at least the supported level before trying to start the queue manager again.

For information about compatibility between releases, see Queue sharing group coexistence on z/OS.

If the results of the MIGRATE QSG option from the CSQ5PQSG utility show queue managers that no longer exist, but are still in the Db2 tables, use the REMOVE QMGR option or, if necessary, the FORCE QMGR option.

CSQ5006E
csect-name Data-sharing groups differ

Explanation

A mismatch has been detected between the Db2 data-sharing group specified on the QSGDATA system parameter and the queue manager entry in the CSQ.ADMIN_B_QMGR table.

System action

The queue manager terminates with completion code X'6C6' and reason code X'00F50006'.

System programmer response

The queue sharing group name specified on the QSGDATA system parameter must match that in which the queue manager is defined in the Db2 CSQ.ADMIN_B_QMGR table.

CSQ5007E
csect-name RRSAF function *function* failed for plan *plan-name*, RC=*return-code* reason=*reason* syncpoint code=*sync-code*

Explanation

A non-zero or unexpected return code was returned from an RRSAF request. The Db2 plan involved was *plan-name*.

System action

If the error occurs during queue manager startup or reconnect processing, the queue manager might terminate with completion code X'6C6' and reason code X'00F50016'. Otherwise, an error message is issued and processing is retried.

System programmer response

Determine the cause of the error using the RRS return and reason code from the message.

See Db2 codes in the *Db2 for z/OS* documentation for an explanation of the codes and attempt to resolve the problem.

CSQ5008E
csect-name Db2 *db2-name* is not a member of data-sharing group *dsg-name*

Explanation

The Db2 subsystem to which the queue manager has connected is not a member of the Db2 data-sharing group specified on the QSGDATA system parameter.

System action

The queue manager terminates with completion code X'6C6' and reason code X'00F50007'.

System programmer response

Ensure that the Db2 subsystem to which the queue manager has connected is a member of the data-sharing group specified on the QSGDATA system parameter.

Issue the Db2 command DIS GROUP to the Db2 subsystem and check the data-sharing group name matches the data-sharing group name on the QSGDATA system parameter.

CSQ5009E
csect-name SQL error for table *table-name*, code=*SQL-code* state=*SQL-state*, data=*d1 d2 d3 d4 d5*

Explanation

A non-zero or unexpected SQL return code was returned from a Db2 SQL request.

System action

The requested operation fails. Processing continues, but the failed request may result in further errors occurring. In some circumstances, the queue manager terminates with completion code X'6C6' and reason code X'00F50014'.

System programmer response

Determine the reason for the SQL error and correct the problem.

See Db2 codes in the *Db2 for z/OS* documentation for an explanation of the SQL error.

CSQ5010E
csect-name XCF IXCQUERY member error, RC=*return-code* reason=*reason*

Explanation

The queue manager received an unexpected return code from an IXCQUERY request.

System action

The queue manager terminates with completion code X'6C6' and reason code X'00F50017'.

System programmer response

Determine the reason for the unexpected error and correct the problem.

Consult the *z/OS MVS Programming: Sysplex Services Reference* manual for an explanation of the return and reason code from the IXCQUERY request.

Ensure that APAR PH64232 has been applied V 10.0.0 or that the queue manager is running at IBM MQ 9.4.3 or later. If not, this message can occur if one or more of the queue managers in a queue sharing group (QSG) do not have a member entry in the XCF group for the QSG. To resolve this issue:

1. Enter the following z/OS command, substituting the QSG name for xxxx:

   ```
   D XCF,GRP,CSQGxxxx,ALL
   ```

   This command lists the members of the XCF group.
2. If any queue managers are defined as a member of the QSG, but do not have an entry in the XCF Group, use the **ADD QMGR** command of the CSQ5PQSG utility to restore the XCF group entry for that queue manager. Run the utility for each queue manager that does not have an entry in the XCF group.

CSQ5011E
csect-name XCF IXCJOIN group error, RC=*return-code* reason=*reason*

Explanation

The queue manager received an unexpected return code from an IXCJOIN request.

System action

The queue manager terminates with completion code X'6C6' and reason code X'00F50019'.

System programmer response

Determine the reason for the unexpected error and correct the problem.

Consult the *z/OS MVS Programming: Sysplex Services Reference* manual for an explanation of the return and reason code from the IXCJOIN request.

CSQ5012E
csect-name XCF IXCQUIES group error, RC=*return-code* reason=*reason*

Explanation

The queue manager received an unexpected return code from an IXCQUIES request.

System action

The queue manager terminates with completion code X'6C6' and reason code X'00F50021'.

System programmer response

Determine the reason for the unexpected error and correct the problem.

Consult the *z/OS MVS Programming: Sysplex Services Reference* manual for an explanation of the return and reason code from the IXCQUIES request.

CSQ5013E
csect-name XCF IXCSETUS error, RC=*return-code* reason=*reason*

Explanation

The queue manager received an unexpected return code from an IXCSETUS request.

System action

The queue manager terminates with completion code X'6C6' and reason code X'00F50018'.

System programmer response

Determine the reason for the unexpected error and correct the problem.

Consult the *z/OS MVS Programming: Sysplex Services Reference* manual for an explanation of the return and reason code from the IXCSETUS request.

CSQ5014I
csect-name Connection to *db2-name* lost, Db2 terminated abnormally

Explanation

The queue manager received an abnormal termination notification from the Db2 subsystem to which it is connected.

System action

The queue manager will clean up its connection to the Db2 subsystem and attempt to reconnect. If a Db2 group attach name was specified on the QSGDATA system parameter a connection to a different Db2 may occur.

System programmer response

Determine the reason for the Db2 abnormal termination. Correct the problem and attempt to restart the Db2 subsystem.

CSQ5015I
csect-name Connection to *db2-name* lost, Db2 shut down forcibly

Explanation

The queue manager received a STOP FORCE termination notification from the Db2 subsystem to which it is connected.

System action

The queue manager will clean up its connection to the Db2 subsystem and attempt to reconnect. If a Db2 group attach name was specified on the QSGDATA system parameter a connection to a different Db2 may occur.

System programmer response

Determine the reason for the Db2 forcible stop. Restart the Db2 subsystem.

CSQ5016I
csect-name Connection to *db2-name* quiescing, Db2 terminating

Explanation

The queue manager received a STOP QUIESCE termination notification from the Db2 subsystem to which it is connected.

System action

The queue manager will quiesce all Db2 server tasks and disconnect from the Db2 subsystem so that it can shut down. It will then attempt to reconnect. If a Db2 group attach name was specified on the QSGDATA system parameter a connection to a different Db2 may occur.

System programmer response

Restart the Db2 subsystem so that shared queue operations can resume.

CSQ5019I
csect-name Disconnected from Db2 *db2-name*

Explanation

The queue manager has successfully disconnected from the Db2 subsystem.

System action

If the disconnect is due to a Db2 STOP MODE(QUIESCE) the queue manager will attempt to reconnect to the Db2 subsystem.

System programmer response

None.

CSQ5020E
csect-name SQL error, table *table-name* not defined in Db2

Explanation

The queue manager attempted to access one of its Db2 tables. Db2 has returned an SQL code indicating the table does not exist.

System action

The request fails and processing continues.

System programmer response

Check that all MQ tasks to set up the Db2 environment completed successfully and that the correct Db2 data-sharing group name was specified on the QSGDATA system parameter.

CSQ5021E
csect-name SQL error, table *table-name* index not built in Db2

Explanation

The queue manager has attempted to access one of its Db2 tables. Db2 has returned an SQL code indicating that the index for the specified table has not been built.

System action

The request fails and processing continues.

System programmer response

Check that all IBM MQ tasks to set up the Db2 environment completed successfully and that the correct Db2 data-sharing group name was specified on the QSGDATA system parameter.

CSQ5022I
csect-name Pending connection to Db2 using *connect-name* ended, queue manager terminating

Explanation

The outstanding connection pending request to Db2 has been terminated due to a STOP QMGR request.

System action

The pending connect to Db2 is canceled and queue manager termination continues.

System programmer response

None.

CSQ5023E
csect-name SQL error, failed to access table *table-name*

Explanation

An attempt by the queue manager to access one of its tables was returned an SQL code indicating that access to the named resource failed.

System action

The request fails and processing continues.

System programmer response

This message will be followed by message CSQ5009E which contains full details of the information returned from Db2 which should be used in conjunction with messages on the Db2 log to diagnose the problem.

The most likely cause of this problem is contention for a Db2 resource, especially on a heavily-used system. If so, the problem is temporary; retry the action that gave the error.

If not, and the problem persists, determine from the message and the Db2 log the resource concerned and perform the recovery actions necessary to unlock the resource. Such a problem could be caused by a Db2 failure while updating one of the Db2 tables, which will be indicated in the Db2 log.

CSQ5024E
csect-name Unable to update queue manager status, RC=*return-code*

Explanation

During startup and shutdown processing the queue manager attempts to update its status in the CSQ.ADMIN_B_QMGR table. This attempt failed.

System action

None. Startup/shutdown processing continues.

System programmer response

None.

CSQ5025E
csect-name SQL error, function *function* code=*SQL-code*

Explanation

A call to the SQL function specified by *function* returned a non-zero code specified by *SQL-code*.

System action

Processing continues.

System programmer response

Note the values contained in the message, and contact IBM Support.

See Db2 codes in the *Db2 for z/OS* documentation for more information about the error code.

CSQ5026E
csect-name Unable to access Db2, RRS is not available

Explanation

The queue manager tried to access Db2, but RRS is not available.

System action

If this occurs during queue manager initialization, the queue manager waits for RRS to become available.

If this occurs at other times, the queue manager terminates its connection to Db2, and then tries to reconnect. Some queue sharing group functions will not be available until RRS is restarted and the connection to Db2 is reestablished.

System programmer response

Start (or restart) RRS.

CSQ5027E
csect-name SQL error for table *table-name*, deadlock or timeout occurred (code=*SQL-code*)

Explanation

An SQL call returned a non-zero code indicating that a deadlock or timeout condition occurred.

System action

The request fails and processing continues.

System programmer response

Retry the command or application involved. If the problem persists, contact your IBM support center.

See Db2 codes in the *Db2 for z/OS* documentation for more information about the error code.

CSQ5028E
csect-name Unable to access Db2, RRS connection limit exceeded

Explanation

The queue manager tried to access Db2, but RRS has reached the limit of allowed concurrent connections (IDENTIFYs).

System action

If this message occurs during queue manager initialization, the queue manager waits for an RRS connection to become available.

If this message occurs at other times, the queue manager terminates its connection to Db2, and then tries to reconnect. Some queue sharing group functions are not available until RRS is restarted and the connection to Db2 is reestablished.

System programmer response

Adjust the RRS connection limit if required, then start (or restart) RRS.

Ensure that the Db2 system parameter controlling the maximum number of concurrent users and connections is correct. The Db2 parameter is Max Batch connect (CTHREAD) on the thread management panel DSNTIPE.

See the *Db2 for z/OS* documentation for an explanation of this Db2 parameter to resolve the problem.

CSQ5029E
csect-name Operation on Db2 table *table-name* failed

Explanation

An operation requested for the named Db2 table failed. For example, the table might be full, or there might be insufficient storage available to perform the request.

This is most likely to occur when writing data to one of the tables that IBM MQ uses to store large shared messages.

System action

Message CSQ5009E is issued giving details of the associated SQL error codes. The requested operation fails and processing continues. The message or other data is not written to the table.

System programmer response

Investigate the cause of the problem as indicated by the SQL codes in message CSQ5009E.

If the table is one of the tables used for storing large shared messages, and the problem is due to insufficient storage, try the operation again later, as the condition might be temporary. If the problem is because the table is full, remove some of the messages; for example, start an application that retrieves and processes the messages. Use the MQ DISPLAY GROUP command to check if there are any obsolete messages in the table space, and delete them. If necessary, increase the size of the table.

CSQ5032I
csect-name Connection to Db2 *db2-name* in data-sharing group *dsg-name* is suspended

Explanation

This is issued in response to a SUSPEND QMGR FACILITY(Db2) command if it completed successfully.

System action

All Db2 activity is suspended for the queue manager named, and the connection to Db2 is broken.

System programmer response

Use the RESUME QMGR FACILITY(Db2) command when ready to resume Db2 activity.

CSQ5033I
csect-name Connection to Db2 *db2-name* in data-sharing group *dsg-name* is resumed

Explanation

The RESUME QMGR FACILITY(Db2) command completed successfully, reestablishing the connection to Db2.

System action

Db2 activity is resumed for the queue manager named.

CSQ5034I
csect-name Suspend or resume Db2 request pending

Explanation

A SUSPEND or RESUME QMGR FACILITY(Db2) command was issued, but such a request is already pending.

System action

None.

System programmer response

Wait until the pending request completes, then reissue the command if necessary.

CSQ5035I
csect-name Connection to Db2 *db2-name* in data-sharing group *dsg-name* already suspended

Explanation

A SUSPEND QMGR FACILITY(Db2) command was issued, but the connection to the named Db2 subsystem is already suspended.

System action

None.

CSQ5036I
csect-name Connection to Db2 *db2-name* in data-sharing group *dsg-name* not suspended

Explanation

A RESUME QMGR FACILITY(Db2) command was issued, but the connection to the named Db2 subsystem is not suspended.

System action

None.

CSQ5038I
csect-name Service task service-task has been unresponsive since hh.mm.ss.nnnnnn. Check for problems with Db2

Explanation

The queue manager has detected a service task *service-task* that is taking too long to process a request that started at hh.mm.ss.nnnnnn.

System action

Processing continues, but certain functions might be unavailable.

System programmer response

Investigate if there are any problems with Db2 or RRS that prevent them responding to IBM MQ requests. For example, the Db2 CTHREAD limit has been exceeded, or Db2 is running slowly because it is short of resources like CPU, I/O capacity, or storage; or Db2 is waiting for log space.

CSQ5039I
csect-name SQL error information

Explanation

An SQL error has occurred. Additional diagnostic information direct from Db2 follows.

System action

See the preceding Db2 manager error message.

System programmer response

Determine the reason for the SQL error and correct the problem.

CSQ5040E
Active log data set xxx.DS01 is encrypted, but active log data set encryption is not supported by all QSG members

Explanation

The named active log data set has been protected by encryption and a queue manager in the same queue sharing group is running at a version of IBM MQ that does not support active log data set encryption.

Active log data set encryption is supported from IBM MQ for z/OS 9.1.4.

Each queue managers in a QSG needs to be able to read the log data sets of every other queue manager in the same QSG

System action

Queue manager start up is abnormally terminated.

System programmer response

Ensure all queue managers in the QSG have been started at a version of IBM MQ that supports active log data set encryption prior to configuring active data set encryption for any queue manager in the QSG.

Either start each queue manager in the QSG at the correct level or reconfigure active logs without data set encryption.

CSQ5041E
csect-name QSG notification flood detected from qmgr-name (outstanding requests=num)

Explanation

The queue manager is receiving notifications from qmgr-name at a faster rate than they can be processed, and the backlog has reached 10000 pending requests.

System action

The message repeats every minute until the backlog from qmgr-name is below 10000.

If the backlog continues to grow, eventual storage exhaustion and abnormal queue manager termination occurs.

System programmer response

Determine which applications on the specified queue manager are responsible for triggering the notifications.

Consider updating the application to use designs that generate fewer notifications.

Examples of application designs that typically generate high numbers of QSG notifications include:

- Using shared application queues that are configured with TRIGTYPE(EVERY).

  Consider using TRIGTYPE(FIRST) instead.
- Applications repeatedly closing the last input handle for a shared queue.

  Consider keeping the handle to the shared queue open until it is no longer required, rather than opening and closing it repeatedly.

V 10.0.0

CSQ5042I
csect-name XCF entry for queue manager qmgr-id (qmgr-name) found.

Explanation

The XCF entry for the queue manager identified was found during startup.

System action

Processing continues.

System programmer response
None

V 10.0.0

CSQ5043I
csect-name XCF entry for queue manager qmgr-id (qmgr-name) recreated.

Explanation

The XCF entry for the queue manager identified was not found during startup.

This is expected when recovering a queue sharing group at an alternative site.

System action

The missing XCF entry is recreated by the starting queue manager.

System programmer response
None

CSQ5100I
DISPLAY GROUP report ...

Explanation

This message is the initial response to the DISPLAY GROUP command. It is followed by message CSQ5102I which is a formatted report of the queue managers in the group.

System action

Processing continues normally.

CSQ5102I
Queue managers in group *group-name*

Explanation

This message is part of the responses to the DISPLAY GROUP command. It provides information about each queue manager in the group, as follows:

```
  Name Num Prefix  Status    Ver  Db2    Connection  name num cpf     qmgr-stat vrm db2-id
conn-stat ⋮  End of queue managers report
```

where:

name
The name of the queue manager.

num
The internally generated number of the queue manager in the group.

cpf
The command prefix of the queue manager.

qmgr-stat
The current status of the queue manager:

ACTIVE
The queue manager is running.

INACTIVE
The queue manager is not running, having terminated normally.

FAILED
The queue manager is not running, having terminated abnormally.

CREATED
The queue manager has been defined to the group, but has not yet been started.

UNKNOWN
The status cannot be determined.

vrm
The function level of the queue manager. The value is a 3-digit number, where:

v
is the version number

r
is the release number

m
is the modification number.

db2-id
The name of the Db2 subsystem or group attachment to which the queue manager connects.

conn-stat
The current status of the connection to Db2:

ACTIVE
The queue manager is running and connected to Db2.

PENDING
The queue manager is running but not connected because Db2 has terminated normally.

FAILED
The queue manager is running but not connected because Db2 has terminated abnormally.

INACTIVE
The queue manager is not running and not connected to Db2.

UNKNOWN
The status cannot be determined.

Exceptionally, the last line might be either:

```
Report terminated, too many lines
```

if the report was generated in response to a command from a z/OS console and more than 253 response lines were generated. Only 253 response lines are returned.

```
Report terminated
```

if there was an error in obtaining the information. The error is described in the following messages.

System action

Processing continues normally.

CSQ5103I
Obsolete messages in Db2 for group *group-name*

Explanation

Messages are normally deleted automatically from Db2, but in exceptional circumstances obsolete messages can remain. This identifies such messages, as follows:

```
  LEID msg-id ⋮  End of messages report
```

where:

msg-id
is the identifier of the message.

Exceptionally, the last line might be either:

```
  Report terminated, too many lines
```

if the report was generated in response to a command from a z/OS console and more than 253 response lines were generated. Only 253 response lines are returned.

```
  Report terminated
```

if there was an error in obtaining the information.

System action

Processing continues normally.

System programmer response

Delete the obsolete messages from Db2. For example, use SPUFI to issue the SQL command

```
  DELETE FROM CSQ.ADMIN_B_MESSAGES     WHERE QSGNAME = 'group-name' AND     LEID = 'msg-
id';
```

See Executing SQL by using SPUFI for more information.

CSQ5113I

Queue manager is not in a queue sharing group

Severity
0

Explanation

A command that requires a queue sharing group was entered, but the queue manager is not in a group.

System action

The command is not actioned.

CSQ5116E

call-name call failed, rc=*rc* reason=*reason*

Severity
8

Explanation

During processing for a DISPLAY GROUP command, a coupling facility services call used to get information failed. *rc* is the return code and *reason* is the reason code (both in hexadecimal) from the call.

System action

Processing is terminated. A following message is issued to identify which type of information was being obtained.

System programmer response

See the *z/OS MVS Programming Sysplex Services Reference.* manual for information about the return and reason codes from the call.

CSQ5117E

Information not available for group *group-name* - reason

Severity

8

Explanation

During processing for a DISPLAY GROUP command, information could not be obtained for the group, for the *reason* indicated:

ERROR
A coupling facility services call failed, as indicated in the preceding CSQ5116E message.

CHANGED
The group size has changed.

System action

Processing is terminated.

System programmer response

Resolve the problem accordingly.

z/OS *Generalized command preprocessor messages (CSQ9...)*

CSQ9000E

'*keyword*' appears more than once

Explanation

The named keyword appears more than once in the command. This message will be issued for each occurrence of the keyword after the first.

System action

Processing for the command is terminated.

System programmer response

Verify the command entry, and reissue the command correctly. See Running MQSC commands from text files for information about the rules for building commands.

CSQ9001E

'*keyword*' is invalid

Explanation

The named keyword is unknown or undefined. It might be misspelled, or it might not be applicable to the command being processed.

System action

Processing for the command is terminated.

System programmer response

Verify the command entry, and reissue the command correctly. See MQSC commands for information about the command.

CSQ9002E

Unbalanced parentheses following '*keyword*'

Explanation

An invalid combination of parentheses has been found following the keyword *keyword*. A closing parenthesis must follow an opening parenthesis before any other opening parenthesis occurs.

System action

Processing for the command is terminated.

System programmer response

Verify the command entry, and reissue the command correctly. See Running MQSC commands from text files for information about the rules for building commands.

CSQ9003E
'*keyword*' parameter contains unbalanced apostrophes

Explanation

An odd number of apostrophes is present in a parameter value of keyword *keyword*. If the parameter is a quoted string, it must have one apostrophe at each end of the string. If an apostrophe is to appear within the string, two adjacent apostrophes must be entered. If the parameter is a hexadecimal value, it must be entered as X'hex-characters'.

System action

Processing for the command is terminated.

System programmer response

Verify the command entry, and reissue the command correctly. See Running MQSC commands from text files for information about the rules for building commands.

CSQ9004E
'*keyword*' parameter specifies range (:) incorrectly

Explanation

A parameter of keyword *keyword* specifies a range of values incorrectly. The character used to denote a range is a colon (:); the format is *lower-limit*:*upper-limit*.

System action

Processing for the command is terminated.

System programmer response

See MQSC commands to verify that the command you are using allows a range for the given keyword. Correct the error, and reissue the command.

CSQ9005E
'*keyword*' parameter does not satisfy generic rules

Explanation

For the keyword *keyword*, parameter values can be generic, but the value specified does not conform to the rules for a generic value. The value does not conform to these rules due to one of the following reasons:

- The value contains an asterisk (*) which is not the last character.
- The value contains a question mark (?) or colon (:).
- The keyword is WHERE and the value is a single asterisk.

System action

Processing for the command is terminated.

System programmer response

Verify the command entry, correct the keyword parameter, and reenter the command. See MQSC commands for a description of the keyword and how to enter the command.

CSQ9006E
'*keyword*' parameter uses asterisk (*) incorrectly

Explanation

For the keyword *keyword*, an asterisk (*) was used in a parameter value. Either:

- The asterisk was not the last or only character in the value. Incorrect examples are NAME(BL*CK) and NAME(*LUE); a correct specification is NAME(BL*) or NAME(*).
- There is a list of parameter values, for example DETAIL(1,*).

System action

Processing for the command is terminated.

System programmer response

See MQSC commands to verify that the command you are using allows specification of '*' for the given keyword. Correct the error, and reissue the command.

CSQ9007E
Either '*keyword1*' or '*keyword2*' must be specified

Explanation

The command requires that either keyword *keyword1* or keyword *keyword2* is specified, but neither keyword was entered on the command. One of the two keywords must be present in order for the command to be processed.

System action

Processing for the command is terminated.

System programmer response

Reissue the command and include whichever keyword is appropriate. See MQSC commands for descriptions of the two keywords. See Running MQSC commands from text files for information about the rules for building commands.

CSQ9008E
'*keyword*' may not be negated

Explanation

The negation characters (NO) appear in front of the keyword *keyword*, but negating this keyword is not allowed.

System action

Processing for the command is terminated.

System programmer response

Verify the command entry, and reissue the command correctly. See Running MQSC commands from text files for further information about this command.

CSQ9009E
'*keyword*' not specified

Explanation

The keyword *keyword* must be present, but it was not entered. This keyword must be present in order for the command to process properly.

System action

Processing for the command is terminated.

System programmer response

Verify the command entry, and reissue the command including the specified keyword. See MQSC commands for further information about this command.

CSQ9010E
Required parameter for '*keyword*' not specified

Explanation

For the keyword *keyword*, either:

- One or more parameters must be specified, but no parameter was entered.
- A fixed number of parameters must be specified, but fewer parameters were entered.

For example, the keyword USERDATA must have a parameter that is a character string. Entering USERDATA() is meaningless; you must either enter a string (for example, USERDATA(MY_DATA)), or if you want to remove this attribute, you must enter USERDATA(' ').

System action

Processing for the command is terminated.

System programmer response

Verify the command entry, supply appropriate parameters for the specified keyword, and reissue the command. See MQSC commands for further information about this command.

CSQ9011E
Parameter(s) not allowed for '*keyword*'

Explanation

No parameters can be specified for the keyword *keyword*. This message is issued for each invalid parameter, so it can be issued more than once for a command.

System action

Processing for the command is terminated.

System programmer response

Verify the command entry, correct the error, and reissue the command. See Running MQSC commands from text files for details on how to enter the command.

CSQ9012E
'*keyword*' parameter is not hexadecimal

Explanation

Parameter values for the keyword *keyword* must be hexadecimal values. Hexadecimal characters are the numeric digits 0 through 9 and the letters A through F, in either uppercase or lowercase. The value can optionally be specified using the hexadecimal string notation X'hex characters'; for example, *keyword*(123ABC) and *keyword*(X'123ABC') are synonymous.

System action

Processing for the command is terminated.

System programmer response

Verify the command entry, and reissue the command, ensuring that the parameters for the named keyword are hexadecimal values.

CSQ9013E
'*keyword*' parameter '*parameter-value*' length is more than *nn*

Explanation

The parameter value *parameter-value* for keyword *keyword* exceeds the limit of *nn* characters in length.

System action

Processing for the command is terminated.

System programmer response

Verify the command entry. See MQSC commands for a list of acceptable parameters. Correct the error, and reissue the command.

CSQ9014E
More than *nn* parameters for '*keyword*'

Explanation

Too many parameters have been specified for the keyword *keyword*. At most *nn* parameters can be specified. In addition to entering too many parameters, this could also be caused by a missing closing parenthesis that has not yet been detected.

System action

Processing for the command is terminated.

System programmer response

Verify the command entry, and reissue the command, using no more than the specified limit of parameters for the given keyword. See MQSC commands for further details. See Running MQSC commands from text files for information about the rules for building commands.

If this error occurs while you are using connection names with the CSQUTIL program you must enclose certain variables within single quotation marks. See CSQUTIL for more information.

CSQ9015E
Parameter '*parameter-value*' is unacceptable for '*keyword*'

Explanation

The parameter value *parameter-value* is not an acceptable value for keyword *keyword*. Either:

- The keyword parameter can be one of a set of character values, but the value specified is not one of them.
- The keyword parameter can be a bounded numeric value, but the value specified is outside the bounds.
- The keyword parameter can be either numeric or one of a set of character values, but the value specified is neither numeric nor one of the set.
- The keyword is WHERE and the first parameter (the filter keyword) is not one of the acceptable keywords for the command.

System action

Processing for the command is terminated.

System programmer response

Verify the command entry, and reissue the command correctly. See MQSC commands for a list of acceptable values. See Running MQSC commands from text files for information about the rules for building commands.

CSQ9016E
'*cmd*' command request not authorized

Explanation

The command requires a level of authorization that you do not have, either for the command itself, or for the resource that it is operating on.

System action

The command is not executed. Processing is terminated.

System programmer response

Contact the system programmer responsible for system security, and request that this person grant you authorization to use the command. Otherwise, you must have someone who is authorized issue the command for you.

CSQ9017E
Failure while processing '*cmd*' command

Explanation

The command preprocessor ended abnormally while processing the command shown in the message. The error is recorded in SYS1.LOGREC, and an SVC dump is requested. The command might have partially completed. Look at any previous response messages to determine what has been done.

System action

Processing for the command is terminated.

System programmer response

Verify the command entry, and reissue the command. If it fails again, collect the items listed in the Problem Determination section, and contact your IBM support center.

CSQ9018E
csect-name Insufficient storage to process '*cmd*' command

Explanation

The command preprocessor was unable to obtain sufficient storage to complete processing of any response messages generated by the command.

System action

Processing for the command is terminated abnormally.

System programmer response

If the problem persists, you might need to increase the region size used by your queue manager or channel initiator, or you might need to reduce the number of jobs running in your system.

CSQ9019E
'*cmd*' command is invalid

Explanation

The command, which starts with *cmd*, is invalid. This could be because:

- the command verb is unknown
- no keywords were specified, or none were specified that are valid as a secondary keyword for the command
- there is syntax error at the start of the command

System action

Processing for the command is terminated.

System programmer response

Verify the command entry, and reissue the command correctly. See MQSC commands for the correct command format. See Running MQSC commands from text files for information about the rules for building commands.

CSQ9020E
'*keyword1*' and '*keyword2*' cannot both be specified

Explanation

The command does not allow keyword *keyword1* and keyword *keyword2* to be specified together.

System action

Processing for the command is terminated.

System programmer response

Verify the command entry, and reissue the command, omitting the inappropriate keyword. See MQSC commands for descriptions of the two keywords. See Running MQSC commands from text files for information about the rules for building commands.

CSQ9022I
csect-name '*cmd*' NORMAL COMPLETION

Explanation

All synchronous processing for the command completed successfully. Any tasks executing asynchronously on behalf of the command might still be executing when this message is displayed.

System action

Synchronous processing for the command is complete.

CSQ9023E
csect-name '*cmd*' ABNORMAL COMPLETION

Explanation

The command has not completed successfully. The command has issued one or more error messages prior to this message.

System action

Processing for the command has ended.

System programmer response

Follow the instructions for the other messages associated with the error.

CSQ9025E
'*parameter-value*' is unacceptable with 'WHERE' parameter '*filter-keyword*'

Explanation

The parameter values for the WHERE keyword are incompatible. The WHERE keyword must have three parameters, *filter-keyword*, *operator*, and *filter-value*. The error is one of the following:

- The operator parameter is not appropriate for the type of parameter values that the filter keyword requires. For example, the filter keyword requires one of a set of parameter values, but the operator is not EQ or NE.
- The filter value parameter exceeds the length limit for parameter values of the filter keyword.
- The filter value parameter is not a value that is valid as a value of the filter keyword. For example:
  - The filter keyword requires a numeric parameter value but the filter value parameter is not numeric.
  - The filter keyword requires one of a set of parameter values but the filter value parameter is not one of them.
  - The filter keyword requires a bounded numeric parameter value but the filter value parameter is outside the bounds.
  - The filter keyword requires an object or system name, but the filter value parameter does not consist only of characters that are valid for such a name.

Depending on the error, *parameter-value* may be the operator parameter or the filter value parameter.

System action

Processing for the command is terminated.

System programmer response

Verify the command entry, and reissue the command correctly. See MQSC commands for information about the parameters for the WHERE keyword.

CSQ9026E
'*keyword*' parameter does not satisfy name rules

Explanation

Parameter values for the keyword *keyword* are names, and therefore must consist only of characters that are valid for the particular type of name, object name or system name. The valid object name characters are uppercase A-Z, lowercase a-z, numerics 0-9, period (.), forward slash (/), underscore (_), and percent sign (%). The valid system name characters are uppercase A-Z, and numerics 0-9; the first character must not be numeric.

This message is issued if the name specified contains invalid characters, or if the name is all blank in cases where an all-blank name is not allowed.

System action

Processing for the command is terminated.

System programmer response

Verify the command entry, and reissue the command ensuring that the parameters for the named keyword are of the required type. See MQSC commands for a description of the keyword. See Running MQSC commands from text files for information about the rules for building commands.

CSQ9028E
'*keyword*' parameter is not numeric

Explanation

Parameter values for the keyword *keyword* must consist of numeric values only.

System action

Processing for the command is terminated.

System programmer response

Verify the command entry, and reissue the command ensuring that the parameters for the named keyword are of the required type. See MQSC commands for a description of the keyword. See Running MQSC commands from text files for information about the rules for building commands.

CSQ9029E
csect-name Failure while processing a command

Explanation

An error occurred while processing a command. The command might or might not have been executed. The error has been recorded in the system error log (the SYS1.LOGREC data set), and an SVC dump was attempted.

You can get this message if you have insufficient ECSA.

System action

Processing for the command is terminated.

System programmer response

Verify the command entry, and reissue the command. If you cannot resolve the problem, collect the items listed in the Problem Determination section, and contact IBM Support.

CSQ9030E
'*keyword*' parameter may not be generic

Explanation

The parameter for the keyword *keyword* specifies a generic value using an asterisk (for example, ABC*), but a generic value is not allowed for that keyword.

System action

Processing for the command is terminated.

System programmer response

Verify the command entry, correct the keyword parameter, and reenter the command. See MQSC commands for a description of the keyword. See Running MQSC commands from text files for information about the rules for building commands.

CSQ9031E
Syntax error following '*keyword*'

Explanation

The text that follows the named keyword contains invalid syntax. This is typically caused by specifying an incorrect sequence of special characters, such as equals (=), comma (,), colon (:), or parentheses.

System action

Processing for the command is terminated.

System programmer response

Verify the command entry, examining the text following the named keyword. Ensure that you have followed the rules for command entry, and reenter the command. Running MQSC commands from text files for information about the rules for building commands.

CSQ9032E
Requested function is not available

Explanation

An attempt was made to invoke a command processor that was not loaded.

System action

The requested function is not performed.

System programmer response

Verify the command entry, to determine which command caused the error.

CSQ9033E

Command exceeds allowable length

Severity
8

Explanation

The command is so large that its internal form has exceeded the maximum length allowed. The size of the internal form of the command is affected by both the length, and the complexity of the command. (For example, an attempt has been made to use the operations and control panels to create a namelist containing too many names.)

This message could also be caused by commands entered through one of the following:

- the initialization input data sets
- the COMMAND function of the utility program CSQUTIL
- a user-written program that puts commands onto the system-command input queue, SYSTEM.COMMAND.INPUT

System action

Processing of the command is terminated.

System programmer response

If you are using the operations and control panels to define a namelist, use the edit facility to reduce the number of names in the list. If you are entering a command from elsewhere, determine which command caused the error, and verify the syntax of that command from MQSC commands. Correct the command.

CSQ9034E

Command cannot be issued using command server

Severity
8

Explanation

An attempt was made to issue a command using the command server. The command cannot be issued in that way.

The command server is used by commands entered through one of the following:

- the COMMAND function of CSQUTIL
- the CSQINPX initialization input data set of the channel initiator
- a user-written program that puts commands onto the system-command input queue, SYSTEM.COMMAND.INPUT

System action

The command is ignored.

CSQ9035E

csect-name Required keyword not specified

Severity
8

Explanation

The command requires one of a set of alternative keywords to be specified, but none was.

System action

Processing for the command is terminated.

System programmer response

Verify the command entry, and reissue the command correctly. See MQSC commands for the proper format of the command. See Running MQSC commands from text files for information about the rules for building commands.

CSQ9036E

Command with '*keyword*(*parameter-value*)' not allowed when queue manager is active

Severity
8

Explanation

The command has the specified parameter value for keyword *keyword*. The command with this keyword and value can be issued only when the queue manager is not active.

System action

The command is ignored.

System programmer response

See MQSC commands for information about how to use the command.

CSQ9037E

Command must be issued from *ddname*

Severity
8

Explanation

An attempt was made to issue a command from the specified initialization input data set. The command cannot be issued from that data set.

System action

The command is ignored.

System programmer response

See MQSC commands for information about how to use the command.

CSQ9038E

Command must be issued from console

Severity
8

Explanation

An attempt was made to issue a command from other than the z/OS console or its equivalent. The command can only be issued in that way.

System action

The command is ignored.

System programmer response

Issue the command from the z/OS console; it cannot be issued from elsewhere.

If you issued the **DEFINE PSID** command from the console, you must include the additional DSN parameter for the command to complete successfully.

See MQSC commands for information about how to use the command.

CSQ9039E

Command cannot be issued from console

Severity
8

Explanation

An attempt was made to issue a command from the z/OS console or its equivalent. The command cannot be issued in that way.

System action

The command is ignored.

System programmer response

See MQSC commands for information about how to use the command.

CSQ9040E

Command cannot be issued from *ddname*

Severity
8

Explanation

An attempt was made to issue a command from the specified initialization input data set. The command cannot be issued from that data set.

System action

The command is ignored.

System programmer response

See MQSC commands for information about how to use the command.

CSQ9041E

Command not allowed during restart

Severity
8

Explanation

An attempt was made to issue a command before restart had completed, but the command cannot be issued at that time. This could be because the command was in the CSQINP1 initialization input data set.

System action

The command is ignored.

System programmer response

If the command was in the CSQINP1 initialization input data set, delete it.

CSQ9042E

Command with '*keyword*()' cannot be issued from *ddname*

Severity
8

Explanation

The command was issued with the specified keyword from an initialization input data set. The command with this keyword cannot be issued from that data set.

System action

The command is ignored.

System programmer response

See MQSC commands for information about how to use the command.

CSQ9045E
'*keyword*' has parameter(s) and is a 'WHERE' parameter

Explanation

The command specifies the WHERE keyword with a filter keyword parameter *keyword*. That keyword is also specified explicitly with with parameters, which is not allowed.

System action

Processing for the command is terminated.

System programmer response

Verify the command entry, and reissue the command correctly. See MQSC commands for information about the parameters for the WHERE keyword.

z/OS IBM MQ for z/OS codes

Each component of IBM MQ for z/OS can issue codes and each component uses a unique two character hexadecimal identifier for its messages. Use this topic to identify and interpret the codes for IBM MQ for z/OS components.

The following code types are described:

z/OS *Connection manager codes (X'94')*

If a connection manager reason code occurs that is not listed here, an internal error has occurred. Collect the following diagnostic items and contact IBM Support.

Diagnostics

- A description of the actions that led to the error or, if applicable, either a listing of the application program or the input string to a utility program that was being run at the time of the error.
- The console output for the period leading up to the error.
- The queue manager job log.
- The system dump resulting from the error.
- If you are using CICS, the CICS transaction dump output.
- Appropriate IBM MQ, z/OS, Db2, CICS, and IMS service levels.
- If you are using the IBM MQ Operations and Control panels, the ISPF panel name.

00940001

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6', and the queue manager terminates.

System programmer response

Collect the items listed in “Diagnostics” on page 943 and contact IBM Support.

Restart your queue manager.

00940003

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 943 and contact IBM Support.

00940004

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 943 and contact IBM Support.

00940007

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 943 and contact IBM Support.

00940008

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6', and the queue manager terminates.

System programmer response

Collect the items listed in “Diagnostics” on page 943 and contact IBM Support.

Restart your queue manager.

00940028

A requested diagnostic trap has occurred.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

This should only occur if IBM Support have requested that a dump be captured to aid in problem diagnosis

Collect the items listed in “Diagnostics” on page 943 and contact IBM Support.

0094002B

An internal error has occurred during ALESERV processing.

System action

The current execution unit terminates with completion code X'5C6'. The failing return code from ALESERV will be in register 2 of the dump.

System programmer response

Collect the items listed in “Diagnostics” on page 943 and contact IBM Support.

Restart the queue manager.

Related tasks

Collecting troubleshooting information

z/OS *Topic Manager codes ('X'A3')*

If a topic manager reason code occurs that is not listed here, an internal error has occurred. Collect the following diagnostic items and contact IBM Support.

Diagnostics

- A description of the actions that led to the error or, if applicable, either a listing of the application program or the input string to a utility program that was being run at the time of the error.
- The console output for the period leading up to the error.
- The queue manager job log.
- The system dump resulting from the error.
- If you are using CICS, the CICS transaction dump output.

- Appropriate IBM MQ, z/OS, Db2, CICS, and IMS service levels.
- If you are using the IBM MQ Operations and Control panels, the ISPF panel name.

00A30001, 00A30002,00A30052, 00A30053, 00A30054, 00A30061, 00A30062, 00A30064, 00A30065, 00A30066, 00A31000

An internal error has occurred while processing a command.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in "Diagnostics" on page 944 and contact IBM Support.

00A30042

An internal error has occurred while processing a command.

If this error occurs in conjunction with a CSQY227E message then the problem is a lack of 64 bit storage.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in "Diagnostics" on page 944 and contact IBM Support.

You should consider raising the value of the MEMLIMIT parameter. For more information, see Address space storage.

00A30072, 00A30073, 00A30074, 00A30075, 00A30076, 00A30077

An internal error occurred during commit processing.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in "Diagnostics" on page 944 and contact IBM Support.

Related tasks

Collecting troubleshooting information

z/OS *Batch adapter codes (X'C2')*

00C20001

The CSQBSRV program has detected a request for a nonexistent function. CSQBSRV is invoked from batch and RRS-batch applications via a stub such as CSQBSTUB, CSQBRRSI, or CSQBRSTB.

System action

The application program ends abnormally, but IBM MQ continues processing.

System programmer response

The most likely cause of this problem is incompatible versions of CSQBSRV and the stub. If this is not the cause of the problem, obtain the diagnostic items listed in this topic, and contact IBM Support.

- Application program listing
- Queue manager job log
- PSW and registers at point of failure

00C20009

The task which started an asynchronous IBM MQ thread (for asynchronous message consumption or asynchronous event listening) has ended before the asynchronous thread which it started had ended. This abend is raised on the asynchronous IBM MQ thread, because processing cannot continue after the resources allocated by the original thread have been released.

System action

The application program ends abnormally, but IBM MQ continues processing.

System programmer response

Ensure that an MQDISC is called for all connections which are used to start asynchronous threads before termination of the task which created the connection.

00C2000A, 00C2000B, 00C2000C, 00C2000D, 00C2000E, 00C2000F

An internal error has occurred while processing an MQCRTMH call.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Obtain the diagnostic items listed in this topic, and contact IBM Support.

- An application program listing.
- The queue manager job log.
- The PSW and registers at point of failure.

Related tasks

Collecting troubleshooting information

z/OS *Coupling Facility codes (X'C5')*

If a coupling facility reason code occurs that is not listed here, an internal error has occurred. Collect the following diagnostic items and contact IBM Support. Restart the queue manager if necessary.

Diagnostics

- A description of the actions that led to the error or, if applicable, either a listing of the application program or the input string to a utility program that was being run at the time of the error.
- The console output for the period leading up to the error.
- The queue manager job log.
- The system dump resulting from the error.
- If you are using CICS, the CICS transaction dump output.
- Appropriate IBM MQ, z/OS, Db2, CICS, and IMS service levels.
- If you are using the IBM MQ Operations and Control panels, the ISPF panel name.
- A dump of the coupling facility structure.

00C50006

A backup or recovery of a CF structure failed because the queue manager is not connected to a Db2 subsystem.

System action

CF structure backup or recovery processing is terminated.

System programmer response

Configure the Db2 subsystem so that the queue manager can connect to it.

00C50012

CF structure processing failed, because the CF structure became full during the action.

System action

CF structure processing is terminated.

System programmer response

Increase the size of the CF structure.

00C50014

An unexpected reason code was returned by the Db2 subsystem that the queue manager is connected to.

System action

The current operation is terminated.

System programmer response

Investigate the cause of the error, as reported in the preceding messages.

00C5004F

This reason code is issued in message CSQM090E when a command has failed. It indicates that a request has been issued for a CF structure, but the request cannot be performed, as explained in the accompanying more specific message.

System action

The command is ignored.

System programmer response

Refer to the description of the accompanying message.

00C50050

The CF structure is being recovered and cannot be used until the recovery is complete.

System action

Processing of the command is terminated.

System programmer response

Wait for the recovery of the structure to complete, then reissue the command. Use the **DISPLAY CFSTATUS** command to view the status of the CF structures.

00C5005B

CF structure recovery failed because an error occurred when reading the BSDS of another queue manager in the queue sharing group.

System action

CF structure recovery processing is terminated.

System programmer response

Check the log for recovery log manager messages that indicate the reason for the error.

00C50064

A backup or recovery of a CF structure failed either because the installation and customization options chosen for IBM MQ do not allow the queue manager to use structures at the required level, or because the level of the structure is not supported by the current command level.

System action

CF structure backup or recovery processing is terminated.

System programmer response

Investigate the cause of the error and ensure that the level of the structure is compatible with the current command level. Where required, adjust the installation and customization options.

00C50D00

A backup of a CF structure failed because a required SMDS data set is not available.

System action

CF structure backup processing is terminated.

System programmer response

Ensure that all SMDS data sets used for the CF structure are available, then reissue the backup command. A **RECOVER CFSTRUCT** command can be used to restore these data sets if this is required.

00C51001, 00C51004, 00C51005, 00C51006, 00C5100A, 00C51019, 00C5101A, 00C5101B, 00C5101C, 00C5001D, 00C5101D, 00C5101E, 00C5101F

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'. In some cases, the queue manager might terminate with completion code X'6C6'.

System programmer response

Collect the items listed in "Diagnostics" on page 946 and contact IBM Support.

Restart the queue manager if necessary.

00C51021, 00C51022, 00C51023, 00C51024, 00C50025, 00C51026, 00C51027, 00C51028, 00C51029, 00C5002A, 00C5102B, 00C5102C, 00C5102D, 00C5102E

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'. In some cases, the queue manager might terminate with completion code X'6C6'.

System programmer response

Collect the items listed in "Diagnostics" on page 946 and contact IBM Support.

Restart the queue manager if necessary.

00C5102F

The queue manager failed to connect to the administration structure.

System action

The current execution unit terminates with completion code X'5C6', and the queue manager attempts to connect to the administration structure if it becomes available again. In some cases, the queue manager might terminate with completion code X'6C6'.

System programmer response

Investigate the configuration for the administration structure. This abend code might be seen if the queue manager is running on an LPAR with no connectivity to the administration structure, or if the administration structure has been encrypted, but the LPAR does not have access to a cryptographic coprocessor.

00C50030, 00C51030, 00C51031, 00C51032, 00C51033, 00C51034, 00C50035, 00C51036, 00C51037, 00C51038, 00C51039, 00C5003A, 00C5103A, 00C5103B, 00C5103C, 00C5103D, 00C5103E, 00C5003F

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'. In some cases, the queue manager might terminate with completion code X'6C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 946 and contact IBM Support.

Restart the queue manager if necessary.

00C50040, 00C51041, 00C51042, 00C51043, 00C51044, 00C50045, 00C51046, 00C51047, 00C51048, 00C51049, 00C5104A

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'. In some cases, the queue manager may terminate with completion code X'6C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 946 and contact IBM Support.

Restart the queue manager if necessary.

00C51050, 00C51051, 00C51052, 00C51053, 00C51054, 00C50055, 00C51056, 00C51057, 00C51058, 00C51059, 00C5105A, 00C5105B

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'. In some cases, the queue manager may terminate with completion code X'6C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 946 and contact IBM Support.

Restart the queue manager if necessary.

00C51060, 00C51061, 00C51062, 00C51063, 00C51064

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'. In some cases, the queue manager might terminate with completion code X'6C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 946 and contact IBM Support.

Restart the queue manager if necessary.

00C51090, 00C51093

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'. In some cases, the queue manager might terminate with completion code X'6C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 946 and contact IBM Support.

Restart the queue manager if necessary.

00C51094, 00C51095, 00C51096, 00C51097

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'. In some cases, the queue manager might terminate with completion code X'6C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 946 and contact IBM Support.

00C510A1, 00C510A2, 00C510A3, 00C510A4, 00C500A5, 00C510A6, 00C510A7, 00C510A8, 00C510A9, 00C500AA

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'. In some cases, the queue manager might terminate with completion code X'6C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 946 and contact IBM Support.

Restart the queue manager if necessary.

00C510AB

The CF structure has failed or connection to it has been lost.

System action

This might be issued in response to a command, in which case processing of the command is terminated. Otherwise, the current execution unit terminates with completion code X'5C6'. In some cases, the queue manager might terminate with completion code X'6C6'.

System programmer response

Restart the queue manager if necessary. Recover the structure; if the error occurred in response to a command, reissue it.

00C510AC, 00C510AD, 00C510AE

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'. In some cases, the queue manager might terminate with completion code X'6C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 946 and contact IBM Support.

Restart the queue manager if necessary.

00C51100, 00C51101, 00C51102, 00C51103, 00C51104, 00C51105, 00C51106, 00C51107, 00C51108, 00C51109, 00C5110A, 00C5110B, 00C5110C, 00C5110D, 00C5110E, 00C5110F

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'. In some cases, the queue manager may terminate with completion code X'6C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 946 and contact IBM Support.

Restart the queue manager if necessary.

00C51110, 00C51111, 00C51112, 00C51113, 00C51114, 00C51115, 00C51116, 00C51117, 00C51118, 00C51119, 00C5111A, 00C5111B, 00C5111C, 00C5111D, 00C5111E, 00C5111F

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'. In some cases, the queue manager might terminate with completion code X'6C6'.

System programmer response

Collect the items listed in "Diagnostics" on page 946 and contact IBM Support.

Restart the queue manager if necessary.

00C51120, 00C51121, 00C51122, 00C51123, 00C51124, 00C51125, 00C51126, 00C51127, 00C51128, 00C51129, 00C5112A, 00C5112B, 00C5112C, 00C5112D, 00C5112E, 00C5112F

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'. In some cases, the queue manager might terminate with completion code X'6C6'.

System programmer response

Collect the items listed in "Diagnostics" on page 946 and contact IBM Support.

Restart the queue manager if necessary.

00C51130, 00C51131, 00C51132, 00C51133, 00C51134, 00C51135, 00C51136, 00C51137, 00C51138, 00C51139, 00C5113A, 00C5113B, 00C5113C, 00C5113D, 00C5113E, 00C5113F

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'. In some cases, the queue manager may terminate with completion code X'6C6'.

System programmer response

Collect the items listed in "Diagnostics" on page 946 and contact IBM Support.

Restart the queue manager if necessary.

00C51140, 00C51141, 00C51142, 00C51143, 00C51144, 00C51145, 00C51146, 00C51147, 00C51148, 00C51149, 00C5114A, 00C5114B, 00C5114C, 00C5114D, 00C5114E, 00C5114F

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'. In some cases, the queue manager may terminate with completion code X'6C6'.

System programmer response

Collect the items listed in "Diagnostics" on page 946 and contact IBM Support.

Restart the queue manager if necessary.

00C51150, 00C51151, 00C51152, 00C51153, 00C51154, 00C51155, 00C51156, 00C51157, 00C51158, 00C51159, 00C5115A, 00C5115B, 00C5115C, 00C5115D, 00C5115E, 00C5115F

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'. In some cases, the queue manager might terminate with completion code X'6C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 946 and contact IBM Support.

Restart the queue manager if necessary.

00C51160, 00C51161, 00C51162, 00C51163, 00C51164, 00C51165, 00C51166, 00C51167, 00C51168, 00C51169, 00C5116A, 00C5116B, 00C5116C, 00C5116D, 00C5116E, 00C5116F

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'. In some cases, the queue manager might terminate with completion code X'6C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 946 and contact IBM Support.

Restart the queue manager if necessary.

00C51170, 00C51171, 00C51172, 00C51174, 00C51175, 00C51176, 00C51177, 00C51178, 00C51179, 00C5117A, 00C5117B, 00C5117C, 00C5117D, 00C5117E, 00C5117F

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'. In some cases, the queue manager might terminate with completion code X'6C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 946 and contact IBM Support.

Restart the queue manager if necessary.

00C51173

An internal error has occurred.

System action

The internal task performing recovery of a CFSTRUCT terminates with completion code x'5C6'.

System programmer response

This error is often, but not exclusively, associated with space issues in the coupling facility.

Ensure that sufficient space is available in the cfstructure.

A common source of error is that the INITSIZE and SIZE values do not match in the CFRM policy. During normal use, the structure has expanded through AUTOALTER processing and the structure backup being restored reflects this size.

However, a new structure has been allocated with the INITSIZE attribute that is too small.

If storage issues are not indicated, then collect the items listed in “Diagnostics” on page 946 and contact IBM Support.

00C51180, 00C51181, 00C51182, 00C51184, 00C51185, 00C51186, 00C51187, 00C51188, 00C51189, 00C5118A, 00C5118B, 00C5118C, 00C5118D, 00C5118E, 00C5118F

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'. In some cases, the queue manager may terminate with completion code X'6C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 946 and contact IBM Support.

Restart the queue manager if necessary.

00C51183

An internal error has occurred.

System action

The internal task performing recovery of a CFSTRUCT terminates with completion code x'5C6'.

System programmer response

This error is often, but not exclusively, associated with space issues in the coupling facility.

Ensure that sufficient space is available in the cfstructure.

A common source of error is that the INITSIZE and SIZE values do not match in the CFRM policy. During normal use, the structure has expanded through AUTOALTER processing and the structure backup being restored reflects this size.

However, a new structure has been allocated with the INITSIZE attribute that is too small.

If storage issues are not indicated, then collect the items listed in “Diagnostics” on page 946 and contact IBM Support.

00C51190, 00C51191, 00C51192, 00C51193, 00C51194, 00C51195, 00C51196, 00C51197, 00C51198, 00C51199, 00C5119A, 00C5119B, 00C5119C, 00C5119D, 00C5119E, 00C5119F

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'. In some cases, the queue manager may terminate with completion code X'6C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 946 and contact IBM Support.

Restart the queue manager if necessary.

00C511A0, 00C511A1, 00C511A2, 00C511A3, 00C511A4, 00C511A5, 00C511A6, 00C511A7, 00C511A8, 00C511A9, 00C511AA, 00C511AB, 00C511AC, 00C511AD, 00C511AE, 00C511AF

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'. In some cases, the queue manager may terminate with completion code X'6C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 946 and contact IBM Support.

Restart the queue manager if necessary.

00C511B0, 00C511B1, 00C511B2, 00C511B3, 00C511B4, 00C511B5, 00C511B6, 00C511B7, 00C511B8, 00C511B9, 00C511BA, 00C511BB, 00C511BC, 00C511BD, 00C511BE, 00C511BF

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'. In some cases, the queue manager may terminate with completion code X'6C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 946 and contact IBM Support.

Restart the queue manager if necessary.

00C511C0, 00C511C1, 00C511C2, 00C511C3, 00C511C4, 00C511C5, 00C511C6, 00C511C7, 00C511C8, 00C511C9, 00C511CA, 00C511CB, 00C511CC, 00C511CD, 00C511CE, 00C511CF

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'. In some cases, the queue manager may terminate with completion code X'6C6'.

System programmer response

Collect the items listed in "Diagnostics" on page 946 and contact IBM Support.

Restart the queue manager if necessary.

00C511D0, 00C511D1, 00C511D2, 00C511D3, 00C511D4, 00C511D5, 00C511D6, 00C511D7, 00C511D8, 00C511D9, 00C511DA, 00C511DB, 00C511DC, 00C511DD, 00C511DE, 00C511DF

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'. In some cases, the queue manager might terminate with completion code X'6C6'.

System programmer response

Collect the items listed in "Diagnostics" on page 946 and contact IBM Support.

Restart the queue manager if necessary.

00C511E0, 00C511E1, 00C511E2, 00C511E3, 00C511E4, 00C511E5, 00C511E6, 00C511E7, 00C511E8, 00C511E9, 00C511EA, 00C511EB, 00C511EC, 00C511ED, 00C511EE, 00C511EF

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'. In some cases, the queue manager might terminate with completion code X'6C6'.

System programmer response

Collect the items listed in "Diagnostics" on page 946 and contact IBM Support.

Restart the queue manager if necessary.

00C511F0, 00C511F1, 00C511F2, 00C511F3, 00C511F4, 00C511F5, 00C511F6, 00C511F7, 00C511F8, 00C511F9, 00C511FA, 00C511FB, 00C511FC, 00C511FD, 00C511FE, 00C511FF

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'. In some cases, the queue manager might terminate with completion code X'6C6'.

System programmer response

Collect the items listed in "Diagnostics" on page 946 and contact IBM Support.

Restart the queue manager if necessary.

00C53000

The queue manager cannot use the administration structure because its size is less than the minimum that IBM MQ requires.

System action

The queue manager terminates with completion code X'6C6'.

System programmer response

Increase the size of the administration structure. See message CSQE022E for more information.

00C53001

The queue manager has detected a mismatch between the queue sharing group creation timestamp in the Db2 tables and the creation timestamp associated with the structure name in message CSQE029E.

System action

The queue manager terminates, a record is written to SYS1.LOGREC and a dump is taken.

System programmer response

Verify the queue manager, queue sharing group and data-sharing group configuration and determine whether a queue manager has configured to connect to a different Db2 data-sharing group.

If the queue manager and queue sharing group configuration is correct then the structure should be deallocated. Having verified that there are only failed-persistent connections remaining to the structure, deallocate it with the z/OS command

```
SETXCF FORCE,STRUCTURE,STRNAME=ext-struc-name
```

(In this command, *ext-struc-name* is formed by prefixing the IBM MQ structure name from message CSQE029E with the queue sharing group name.)

00C53002

The queue manager cannot use the administration structure because the administration structure is full and remains full despite repeated attempts to wait for space to become available.

System action

The queue manager terminates with completion code X'5C6'.

System programmer response

Increase the size of the administration structure. See message CSQE038E for more information.

Related tasks

Collecting troubleshooting information

z/OS *Message generator codes (X'C6')*

00C60001

IBM MQ received return code X'20' when issuing a WTO request to display a console message. This means that there are no message buffers for either Multiple Console Support (MCS) or JES3, or there is a JES3 WTO staging area excess. The WTO request is terminated. The current console message and all subsequent informational console messages are ignored until the problem is corrected.

System action

A record is written to SYS1.LOGREC. A retry is requested and execution continues. IBM MQ resumes issuing console messages when the condition is corrected.

00C60004

The queue manager was unable to load the message table (CSQFMTAB).

System action

The queue manager terminates.

System programmer response

Ensure that the message table is in the required library (SCSQANLx, where x is your national language letter), that it is referenced correctly, and that all the libraries in the concatenation are APF authorized. Restart the queue manager.

00C60005

An internal error has occurred.

System action

The queue manager is terminated, and a dump is produced.

System programmer response

Restart the queue manager.

Collect the following diagnostic items and contact IBM Support:

- Queue manager job log
- System dump resulting from the error

00C60006

The MQ utility program was unable to load its message table (CSQFSTAB).

System action

The utility program ends abnormally.

System programmer response

Check the console for messages indicating why CSQFSTAB was not loaded. Ensure that the message table is in the required library (SCSQANLx, where x is your national language letter), and that it is referenced correctly, and resubmit the job.

The utility program attempts to load this module from the library data sets under the STEPLIB DD statement of the utility address space.

00C60007

The IBM MQ CICS adapter was unable to load its message table (CSQFCTAB).

System action

The IBM MQ CICS adapter server task terminates.

System programmer response

Check the console for messages indicating why CSQFCTAB was not loaded. Ensure that the message table is in the required library (SCSQANLx, where x is your national language letter), and that it is referenced correctly.

CSQCSERV attempts to load this module from the library data sets under the STEPLIB DD statement of the CICS address space.

00C60008

The IBM MQ utility program was unable to load its message table (CSQFLTAB).

System action

The utility program ends abnormally.

System programmer response

Check the console for messages indicating why CSQFLTAB was not loaded. Ensure that the message table is in the required library (SCSQANLx, where x is your national language letter), and that it is referenced correctly, and resubmit the job.

The utility program attempts to load this module from the library data sets under the STEPLIB DD statement of the utility address space.

00C6000A

The IBM MQ early processing program was unable to load its message table (CSQ3ECMX).

System action

The queue manager terminates.

System programmer response

Ensure that the message table is in the required library (SCSQSNLx, where x is your national language letter) and that it is referenced correctly, and perform an IPL of your z/OS system or use the z/OS command SETSSI ADD to restart the queue manager.

00C6000B

The distributed queuing component was unable to load its message table (CSQFXTAB).

System action

The channel initiator ends.

System programmer response

Check the console for messages indicating why CSQFXTAB was not loaded. Ensure that the message table is in the required library (SCSQANLx, where x is your national language letter), that it is referenced correctly, and that all the libraries in the concatenation are APF authorized. Restart the channel initiator.

00C6000C

The IMS trigger monitor was unable to load its message table (CSQFSTAB).

System action

The trigger monitor ends.

System programmer response

Check the console for messages indicating why CSQFSTAB was not loaded. Ensure that the message table is in the required library (SCSQANLx, where x is your national language letter), and that it is referenced correctly, and restart the trigger monitor.

00C600F0

The Advanced Message Security component was unable to load its message table (CSQF0TAB).

System action

The Advanced Message Security component fails to start during queue manager startup.

System programmer response

Check the console for messages indicating why CSQF0TAB was not loaded. Ensure that the message table is in the required library (SCSQANLx, where x is your national language letter), that it is referenced correctly, and that all the libraries in the concatenation are APF authorized. Restart the queue manager.

Related tasks

Collecting troubleshooting information

z/OS *Functional recovery manager codes (X'C7')*

00C70010

While trying to recover from an error, an internal consistency check indicated a storage overlay, or an internal error.

System action

Control is percolated to the z/OS recovery termination manager, and a dump is requested.

System programmer response

Retain the dump, and contact IBM Support for assistance.

Restart the queue manager if necessary.

00C70020

A critical procedure recovery routine has ended abnormally, causing a secondary abnormal end.

System action

Control is percolated to the z/OS recovery termination manager, and in some cases the queue manager terminates abnormally. A dump is produced for both the primary and secondary errors.

System programmer response

Retain both dumps, and contact IBM Support for assistance.

Restart the queue manager if necessary.

00C70030

A request to z/OS to establish an ESTAE produced a non-zero return code.

System action

A dump is requested.

System programmer response

The return code from z/OS is captured in register 14.

Go to the z/OSMVS Programming: Assembler Services Reference, Volume 1 (ABEND-HSPSERV) manual for an explanation of the return code.

00C70040

This abnormal end reason code was caused by an internal IBM MQ error.

System action

Control is percolated to the z/OS recovery termination manager, and a dump is requested.

System programmer response

Retain the dump, and contact IBM Support for assistance.

Restart the queue manager if necessary.

Related tasks

Collecting troubleshooting information

z/OS *Security manager codes (X'C8')*

If a security manager reason code occurs that is not listed here, an internal error has occurred. Collect the following diagnostic items and contact IBM Support.

Diagnostics

- A description of the actions that led to the error or, if applicable, either a listing of the application program or the input string to a utility program that was being run at the time of the error.
- The console output for the period leading up to the error.
- The queue manager job log.
- The system dump resulting from the error.

- If you are using CICS, the CICS transaction dump output.
- Appropriate IBM MQ, z/OS, Db2, CICS, and IMS service levels.
- If you are using the IBM MQ Operations and Control panels, the ISPF panel name.
- The security command issued before the error.

00C80001

An attempt to obtain storage for the security manager was unsuccessful.

Note: This could indicate a system-wide storage problem.

System action

The queue manager is terminated, and a dump is produced. Register 2 contains the return code from the storage failure.

System programmer response

Check that you are running with the recommended region size, and if not, reset your system and restart the queue manager. If this is not the cause of the problem, use these items to diagnose the cause of the problem:

- Queue manager job log
- Information about any other storage-related problems
- System dump resulting from the error

00C80002

An attempt to obtain storage for the security manager was unsuccessful.

Note: This error code could indicate a system-wide storage problem.

System action

The queue manager is terminated, and a dump is produced. Register 2 contains the return code from the storage failure.

System programmer response

Check that you are running with the suggested region size, and if not, reset your system and restart the queue manager. If this is not the cause of the problem, use these items to diagnose the cause of the problem:

- Queue manager job log
- Information about any other storage-related problems
- System dump resulting from the error

00C80003

An attempt to obtain a storage subpool for the security manager was unsuccessful.

Note: This error code could indicate a system-wide storage problem.

System action

The queue manager is terminated, and a dump is produced. Register 2 contains the return code from the storage failure.

System programmer response

Check that you are running with an appropriate region size, and if not, reset your system and restart the queue manager. If the region size is not the cause of the problem, use these items to diagnose the cause of the problem:

- Queue manager job log
- Information about any other storage-related problems
- System dump resulting from the error

00C80004

An internal error has occurred.

System action

The queue manager is terminated, and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 958 and contact IBM Support.

Restart the queue manager.

00C8000A

A severe error has occurred during a SAF RACROUTE REQUEST=STAT call to the external security manager (ESM) during security switch processing at queue manager initialization time.

System action

The queue manager is terminated, and a dump is produced. Register 2 contains the address of the class being checked at the time of the error.

System programmer response

See your ESM documentation for information about any return codes that appear in the job log. Check your security configuration (for example, that the required classes are installed and active). If you are unable to resolve the problem, collect the items listed in “Diagnostics” on page 958 and contact IBM Support.

00C8000B

A severe error has occurred during a SAF RACROUTE REQUEST=EXTRACT call to the external security manager (ESM) during security switch processing at queue manager initialization time.

System action

The queue manager is terminated, and a dump is produced. Register 2 contains the address of the entity being checked at the time of the error.

System programmer response

See your ESM documentation for information about any return codes that appear in the job log. For information about setting IBM MQ security switches, see Switch profiles. If you are unable to resolve the problem, collect the items listed in “Diagnostics” on page 958 and contact IBM Support.

00C8000C

A severe error has occurred during a SAF RACROUTE REQUEST=LIST (create) call to the external security manager (ESM) during security switch processing at queue manager initialization time.

System action

The queue manager is terminated, and a dump is produced. Register 2 contains the address of the class, and register 3 the address of the entity, being checked at the time of the error.

System programmer response

See your ESM documentation for information about any return codes that appear in the job log. If you are unable to resolve the problem, collect the items listed in “Diagnostics” on page 958 and contact IBM Support.

00C8000D

An unexpected return code has been received from one of the following SAF calls to the external security manager (ESM) during security switch processing at queue manager initialization time:

- RACROUTE REQUEST=EXTRACT
- RACROUTE REQUEST=LIST

- RACROUTE REQUEST=STAT

System action

Message CSQH004I is produced containing the return codes from SAF and the ESM. The queue manager is terminated, and a dump is produced. Register 2 contains the address of the return codes.

System programmer response

See your ESM documentation for information about the return codes that appear in message CSQH004I (in the job log) or the dump. For information about setting IBM MQ security switches, see Switch profiles. If you are unable to resolve the problem, collect the items listed in "Diagnostics" on page 958 and contact IBM Support.

00C8000E

An unexpected setting for the subsystem security switch was encountered.

System action

The queue manager is terminated, and a dump is produced. Register 2 contains the address of the control block containing the switch setting.

System programmer response

Collect the items listed in "Diagnostics" on page 958, together with a note of what you expected the switch to be set to, and whether you had defined a profile for it or not, and contact IBM Support.

00C8000F

An internal error has occurred.

System action

The queue manager is terminated, and a dump is produced. Register 2 contains the address of the class involved at the time of the error.

System programmer response

Collect the items listed in "Diagnostics" on page 958 and contact IBM Support.

Restart the queue manager.

00C80010

An attempt to obtain storage for the security manager was unsuccessful.

Note: This error code could indicate a system-wide storage problem.

System action

The current execution unit terminates with a completion code of X'5C6', and a dump is produced. Register 2 contains the return code from the storage failure.

System programmer response

Check that you are running with the suggested region size, and if not, reset your system and restart the queue manager. If this is not the cause of the problem, use the items listed in "Diagnostics" on page 958, together with information about any other storage-related problems, to diagnose the cause of the problem. If you are unable to resolve the problem, contact IBM Support.

00C80011

An attempt to obtain a storage subpool for the security manager was unsuccessful.

Note: This error code could indicate a system-wide storage problem.

System action

The current execution unit terminates with a completion code of X'5C6', and a dump is produced. Register 2 contains the return code from the storage failure.

System programmer response

Check that you are running with the suggested region size, and if not, reset your system and restart the queue manager. If this is not the cause of the problem, use the items listed in “Diagnostics” on page 958, together with information about any other storage-related problems, to diagnose the cause of the problem. If you are unable to resolve the problem, contact IBM Support.

00C80012

An attempt to obtain storage for the security manager was unsuccessful.

Note: This error code could indicate a system-wide storage problem.

System action

The current execution unit terminates with a completion code of X'5C6', and a dump is produced. Register 2 contains the return code from the storage failure.

System programmer response

Check that you are running with the suggested region size, and if not, reset your system and restart the queue manager. If this is not the cause of the problem, use the items listed in “Diagnostics” on page 958, together with information about any other storage-related problems, to diagnose the cause of the problem. If you are unable to resolve the problem, contact IBM Support.

00C80013

An internal error has occurred while processing a security request.

System action

The current execution unit terminates with a completion code of X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 958 and contact IBM Support.

00C80020

An attempt to obtain storage for the security manager was unsuccessful.

Note: This error code could indicate a system-wide storage problem.

System action

The current execution unit terminates with a completion code of X'5C6', and a dump is produced. Register 2 contains the return code from the storage failure.

System programmer response

Check that you are running with the suggested region size, and if not, reset your system and restart the queue manager. If this is not the cause of the problem, use the items listed in “Diagnostics” on page 958, together with information about any other storage-related problems, to diagnose the cause of the problem. If you are unable to resolve the problem, contact IBM Support.

00C80024

An internal error has occurred while processing a command.

System action

The current execution unit terminates with a completion code of X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 958 and contact IBM Support.

00C80025

An internal error has occurred while processing a command.

System action

The current execution unit terminates with a completion code of X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 958 and contact IBM Support.

00C80026

An internal error has occurred while processing a command.

System action

The current execution unit terminates with a completion code of X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 958 and contact IBM Support.

00C80027

An unrecognized keyword was encountered whilst processing a REFRESH SECURITY command.

System action

The current execution unit terminates with a completion code of X'5C6', and a dump is produced. Register 2 contains the address of the keyword causing the problem.

System programmer response

Collect the items listed in “Diagnostics” on page 958 and contact IBM Support.

00C80028

An attempt to obtain a storage subpool for the security manager was unsuccessful. This might have occurred during the processing of an ALTER SECURITY command, a REFRESH SECURITY command, or during the automatic security timeout processing.

Note: This could indicate a system-wide storage problem.

System action

The current execution unit terminates with a completion code of X'5C6', and a dump is produced. Register 2 contains the return code from the storage failure.

System programmer response

Use the items listed in “Diagnostics” on page 958, together with information about any other storage-related problems, to diagnose the cause of the problem. If you are unable to resolve the problem, contact IBM Support.

00C80029

A severe error has occurred during a SAF RACROUTE REQUEST=STAT call to the external security manager (ESM) during security switch processing for a REFRESH SECURITY command.

System action

The current execution unit terminates with a completion code of X'5C6', and a dump is produced. Register 2 contains the address of the class being checked at the time of the error.

System programmer response

See your ESM documentation for information about any return codes that appear in the job log. Check your security configuration (for example, that the required classes are installed and active). If you are unable to resolve the problem, collect the items listed in “Diagnostics” on page 958 and contact IBM Support.

00C80031

A severe error has occurred during a SAF RACROUTE REQUEST=LIST (create) call to the external security manager (ESM) during the processing for a REFRESH SECURITY command.

System action

The current execution unit terminates with a completion code of X'5C6', and a dump is produced. Register 2 contains the address of the class, and register 3 the address of the entity, being checked at the time of the error.

System programmer response

See your ESM documentation for information about any return codes that appear in the job log. Check your security configuration (for example, that the required classes are installed and active). If you are unable to resolve the problem, collect the items listed in "Diagnostics" on page 958 and contact IBM Support.

00C80032

An unexpected return code has been received from one of the following SAF calls to the external security manager (ESM) during the processing of a REFRESH SECURITY command:

- RACROUTE REQUEST=LIST (create)
- RACROUTE REQUEST=LIST (delete)
- RACROUTE REQUEST=STAT

System action

The current execution unit terminates with a completion code of X'5C6', and a dump is produced. Register 2 contains the address of the return codes from SAF, and the ESM.

Note: If the error occurred on a STAT call, the error is preceded by a CSQH004I message containing the return codes from SAF, and the ESM.

System programmer response

See your ESM documentation for information about the return codes from SAF and the ESM. For information about setting IBM MQ security switches, see Switch profiles. If you are unable to resolve the problem, collect the items listed in "Diagnostics" on page 958 and contact IBM Support.

00C80033

An unexpected setting for the subsystem security switch was encountered during the processing of a REFRESH SECURITY command.

System action

The current execution unit terminates with a completion code of X'5C6', and a dump is produced.

System programmer response

Collect the items listed in "Diagnostics" on page 958, together with a note of what you expected the switch to be set to, and whether you had defined a profile for it or not, and contact IBM Support.

00C80034

An internal error has occurred.

System action

The current execution unit terminates with a completion code of X'5C6', and a dump is produced. Register 2 contains the address of the class invoked at the time of the check.

System programmer response

Collect the items listed in "Diagnostics" on page 958 and contact IBM Support.

00C80035

A severe error has occurred during a SAF RACROUTE REQUEST=STAT call to the external security manager (ESM) during security switch processing for a REFRESH SECURITY command.

System action

The current execution unit terminates with a completion code of X'5C6', and a dump is produced. Register 2 contains the address of the class being checked at the time of the error.

System programmer response

See your ESM documentation for information about any return codes that appear in the job log. Check your security configuration (for example, that the required classes are installed and active). If you are unable to resolve the problem, collect the items listed in “Diagnostics” on page 958 and contact IBM Support.

00C80036

A severe error has occurred during a SAF RACROUTE REQUEST=EXTRACT call to the external security manager (ESM) during security switch processing for a REFRESH SECURITY command.

System action

The current execution unit terminates with a completion code of X'5C6', and a dump is produced. Register 2 contains the address of the entity being checked at the time of the error.

System programmer response

See your ESM documentation for information about any return codes that appear in the job log. For information about setting IBM MQ security switches, see Switch profiles. If you are unable to resolve the problem, collect the items listed in “Diagnostics” on page 958 and contact IBM Support.

00C80037

A severe error has occurred during a SAF RACROUTE REQUEST=LIST (create) call to the external security manager (ESM) during the processing for a REFRESH SECURITY command.

System action

The current execution unit terminates with a completion code of X'5C6', and a dump is produced. Register 2 contains the address of the class, and register 3 the address of the entity, being checked at the time of the error.

System programmer response

See your ESM documentation for information about any return codes that appear in the job log. If you are unable to resolve the problem, collect the items listed in “Diagnostics” on page 958 and contact IBM Support.

00C80038

An unexpected return code has been received from one of the following SAF calls to the external security manager (ESM) during the processing of a REFRESH SECURITY command.

- RACROUTE REQUEST=LIST (create)
- RACROUTE REQUEST=LIST (delete)
- RACROUTE REQUEST=EXTRACT
- RACROUTE REQUEST=STAT

System action

The current execution unit terminates with a completion code of X'5C6', and a dump is produced. Register 2 contains the address of the return codes from SAF, and the ESM.

Note: If the error occurred on a STAT call, the error is preceded by a CSQH004I message containing the return codes from SAF, and the ESM.

System programmer response

See your ESM documentation for information about the return codes from SAF and the ESM. For information about setting IBM MQ security switches, see Switch profiles. If you are unable to resolve the problem, collect the items listed in “Diagnostics” on page 958 and contact IBM Support.

00C80039

An attempt to obtain a storage subpool for a security manager user entry block was unsuccessful. This could have occurred during either security timeout processing, or REFRESH SECURITY command processing.

Note: This could indicate a system-wide storage problem.

System action

The current execution unit terminates with a completion code of X'5C6', and a dump is produced. Register 2 contains the return code from the storage failure.

System programmer response

Use the items listed in "Diagnostics" on page 958, together with information about any other storage-related problems, to diagnose the cause of the problem. If you are unable to resolve the problem, contact IBM Support.

00C80040

A severe error has occurred during security timeout processing. An unexpected return code has been received from the IBM MQ timer component.

Note: This could indicate a system-wide problem with the timer component, or the system timer.

System action

Messages CSQH009I and CSQH010I are issued. The current execution unit terminates with a completion code of X'5C6', and a dump is produced. Register 2 contains the return code from the timer component that caused the problem.

System programmer response

Use the items listed in "Diagnostics" on page 958, together with information about any other timer-related problems, to diagnose the cause of the problem. If you are unable to resolve the problem, contact IBM Support.

00C80041

A severe error has occurred during security timeout processing for an ALTER SECURITY command. An unexpected return code has been received from the IBM MQ timer component.

Note: This could indicate a system-wide problem with the timer component, or the system timer.

System action

Message CSQH010I is issued. The current execution unit terminates with a completion code of X'5C6' and a dump is produced. Register 2 contains the return code from the timer component that caused the problem.

System programmer response

Use the items listed in "Diagnostics" on page 958, together with information about any other timer-related problems, to diagnose the cause of the problem. If you are unable to resolve the problem, contact IBM Support.

00C80042

A severe error has occurred during security initialization when trying to start the security timer. An unexpected return code has been received from the IBM MQ timer component.

Note: This could indicate a system-wide problem with the timer component, or the system timer.

System action

Message CSQH010I is issued. The queue manager terminates and a dump is produced. Register 2 contains the return code from the timer component that caused the problem.

System programmer response

Use the items listed in “Diagnostics” on page 958, together with information about any other timer-related problems, to diagnose the cause of the problem. If you are unable to resolve the problem, contact IBM Support.

00C80043

A severe error has occurred whilst processing a DISPLAY SECURITY command. A parameter has been entered on the SECURITY keyword, but this is invalid.

System action

The current execution unit terminates with a completion code of X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 958 and contact IBM Support.

00C80044

A severe error has occurred whilst processing an ALTER SECURITY command. A parameter has been entered on the SECURITY keyword, but this is invalid.

System action

The current execution unit terminates with a completion code of X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 958 and contact IBM Support.

00C80045

A severe error has occurred because the last security refresh did not complete successfully.

System action

The current execution unit terminates with error reason code X'5C6', and a dump is produced.

System programmer response

If you are able to fix the cause of the problem, you must refresh the security again before you can continue. If you are unable to solve the problem, collect the items listed in “Diagnostics” on page 958 and contact IBM Support.

00C80046

An attempt to obtain a storage subpool for the security manager Utoken blocks was unsuccessful.

This indicates that there could be a wider ranging problem relating to storage availability.

System action

The queue manager is terminated and a dump is produced.

System programmer response

Use the items listed in “Diagnostics” on page 958, together with information about any other storage-related problems, to diagnose the cause of the problem.

00C80047

An attempt to obtain a storage block for a security manager Utoken block was unsuccessful.

This indicates that there could be a wider ranging problem relating to storage availability.

System action

The current execution unit terminates with X'5C6' and a dump is produced.

System programmer response

Use the items listed in “Diagnostics” on page 958, together with information about any other storage-related problems, to diagnose the cause of the problem. If you are unable to resolve the problem, contact IBM Support.

00C80050

A severe error has occurred during a SAF RACROUTE REQUEST=STAT call to the External Security Manager (ESM) during security switch processing at queue manager initialization time.

System action

The queue manager is terminated, and a dump is produced. Register 2 contains the address of the class being checked at the time of the error.

System programmer response

See your ESM documentation for information about any return codes that appear in the job log. Check your security configuration (for example, that the required classes are installed and active). If you are unable to resolve the problem, collect the items listed in “Diagnostics” on page 958 and contact IBM Support.

00C80051

A severe error has occurred during a SAF RACROUTE REQUEST=EXTRACT call to the External Security Manager (ESM) during security switch processing at queue manager initialization time.

System action

The queue manager is terminated, and a dump is produced. Register 2 contains the address of the entity being checked at the time of the error.

System programmer response

See your ESM documentation for information about any return codes that appear in the job log. For information about setting IBM MQ security switches, see Switch profiles. If you are unable to resolve the problem, collect the items listed in “Diagnostics” on page 958 and contact IBM Support.

00C80052

A severe error has occurred during a SAF RACROUTE REQUEST=LIST (create) call to the External Security Manager (ESM) during security switch processing at queue manager initialization time.

System action

The queue manager is terminated, and a dump is produced. Register 2 contains the address of the class, and register 3 the address of the entity, being checked at the time of the error.

System programmer response

See your ESM documentation for information about any return codes that appear in the job log. If you are unable to resolve the problem, collect the items listed in “Diagnostics” on page 958 and contact IBM Support.

00C80053

An unexpected return code has been received from one of the following SAF calls to the External Security Manager (ESM) during security switch processing at queue manager initialization time.

- RACROUTE REQUEST=EXTRACT
- RACROUTE REQUEST=LIST
- RACROUTE REQUEST=STAT

System action

Message CSQH004I is produced containing the return codes from SAF and the ESM. The queue manager is terminated, and a dump is produced. Register 2 contains the address of the return codes.

System programmer response

See your ESM documentation for information about the return codes that appear in message CSQH004I (in the job log) or the dump. For information about setting IBM MQ security switches, see Switch profiles. If you are unable to resolve the problem, collect the items listed in "Diagnostics" on page 958 and contact IBM Support.

00C80054

An unexpected setting for the subsystem security switch was encountered.

System action

The queue manager is terminated, and a dump is produced. Register 2 contains the address of the control block containing the switch setting.

System programmer response

Collect the items listed in "Diagnostics" on page 958, together with a note of what you expected the switch to be set to, and whether you had defined a profile for it or not, and contact IBM Support.

Restart the queue manager.

00C80055

An internal loop count was exceeded during security switch processing at queue manager initialization time.

System action

The queue manager is terminated, and a dump is produced. Register 2 contains the address of the class being checked at the time of the error.

System programmer response

Collect the items listed in "Diagnostics" on page 958 and contact IBM Support.

Restart the queue manager.

00C80060

A severe error has occurred during a SAF RACROUTE REQUEST=STAT call to the External Security Manager (ESM) during security switch processing at queue manager initialization time.

System action

The queue manager is terminated, and a dump is produced. Register 2 contains the address of the class being checked at the time of the error.

System programmer response

See your ESM documentation for information about any return codes that appear in the job log. Check your security configuration (for example, that the required classes are installed and active). If you are unable to resolve the problem, collect the items listed in "Diagnostics" on page 958 and contact IBM Support.

00C80061

A severe error has occurred during a SAF RACROUTE REQUEST=EXTRACT call to the External Security Manager (ESM) during security switch processing at queue manager initialization time.

System action

The queue manager is terminated, and a dump is produced. Register 2 contains the address of the entity being checked at the time of the error.

System programmer response

See your ESM documentation for information about any return codes that appear in the job log. For information about setting IBM MQ security switches, see Switch profiles. If you are unable to resolve the problem, collect the items listed in "Diagnostics" on page 958 and contact IBM Support.

00C80062

A severe error has occurred during a SAF RACROUTE REQUEST=LIST (create) call to the External Security Manager (ESM) during security switch processing at queue manager initialization time.

System action

The queue manager is terminated, and a dump is produced. Register 2 contains the address of the class, and register 3 the address of the entity, being checked at the time of the error.

System programmer response

See your ESM documentation for information about any return codes that appear in the job log. If you are unable to resolve the problem, collect the items listed in "Diagnostics" on page 958 and contact IBM Support.

00C80063

An unexpected return code has been received from one of the following SAF calls to the External Security Manager (ESM) during security switch processing at queue manager initialization time.

- RACROUTE REQUEST=EXTRACT
- RACROUTE REQUEST=LIST
- RACROUTE REQUEST=STAT

System action

Message CSQH004I is produced containing the return codes from SAF and the ESM. The queue manager is terminated, and a dump is produced. Register 2 contains the address of the return codes.

System programmer response

See your ESM documentation for information about the return codes that appear in message CSQH004I (in the job log) or the dump. For information about setting IBM MQ security switches, see Switch profiles. If you are unable to resolve the problem, collect the items listed in "Diagnostics" on page 958 and contact IBM Support.

00C80064

An unexpected setting for the subsystem security switch was encountered.

System action

The queue manager is terminated, and a dump is produced. Register 2 contains the address of the control block containing the switch setting.

System programmer response

Collect the items listed in "Diagnostics" on page 958, together with a note of what you expected the switch to be set to, and whether you had defined a profile for it or not, and contact IBM Support.

Restart the queue manager.

00C80065

An internal loop count was exceeded during security switch processing at queue manager initialization time.

System action

The queue manager is terminated, and a dump is produced. Register 2 contains the address of the class being checked at the time of the error.

System programmer response

Collect the items listed in "Diagnostics" on page 958 and contact IBM Support.

Restart the queue manager.

00C80070

A severe error has occurred during a SAF RACROUTE REQUEST=STAT call to the External Security Manager (ESM) during security switch processing at queue manager initialization time.

System action

The queue manager is terminated, and a dump is produced. Register 2 contains the address of the class being checked at the time of the error.

System programmer response

See your ESM documentation for information about any return codes that appear in the job log. Check your security configuration (for example, that the required classes are installed and active). If you are unable to resolve the problem, collect the items listed in "Diagnostics" on page 958 and contact IBM Support.

00C80071

A severe error has occurred during a SAF RACROUTE REQUEST=EXTRACT call to the External Security Manager (ESM) during security switch processing at queue manager initialization time.

System action

The queue manager is terminated, and a dump is produced. Register 2 contains the address of the entity being checked at the time of the error.

System programmer response

See your ESM documentation for information about any return codes that appear in the job log. For information about setting IBM MQ security switches, see Switch profiles. If you are unable to resolve the problem, collect the items listed in "Diagnostics" on page 958 and contact IBM Support.

00C80072

A severe error has occurred during a SAF RACROUTE REQUEST=LIST (create) call to the External Security Manager (ESM) during security switch processing at queue manager initialization time.

System action

The queue manager is terminated, and a dump is produced. Register 2 contains the address of the class, and register 3 the address of the entity, being checked at the time of the error.

System programmer response

See your ESM documentation for information about any return codes that appear in the job log. If you are unable to resolve the problem, collect the items listed in "Diagnostics" on page 958 and contact IBM Support.

00C80073

An unexpected return code has been received from one of the following SAF calls to the External Security Manager (ESM) during security switch processing at queue manager initialization time.

- RACROUTE REQUEST=EXTRACT
- RACROUTE REQUEST=LIST
- RACROUTE REQUEST=STAT

System action

Message CSQH004I is produced containing the return codes from SAF and the ESM. The queue manager is terminated, and a dump is produced. Register 2 contains the address of the return codes.

System programmer response

See your ESM documentation for information about the return codes that appear in message CSQH004I (in the job log) or the dump. For information about setting IBM MQ security switches, see Switch profiles. If you are unable to resolve the problem, collect the items listed in "Diagnostics" on page 958 and contact IBM Support.

00C80074

An unexpected setting for the subsystem security switch was encountered.

System action

The queue manager is terminated, and a dump is produced. Register 2 contains the address of the control block containing the switch setting.

System programmer response

Collect the items listed in "Diagnostics" on page 958, together with a note of what you expected the switch to be set to, and whether you had defined a profile for it or not, and contact IBM Support.

00C80075

An internal loop count was exceeded during security switch processing at queue manager initialization time.

System action

The queue manager is terminated, and a dump is produced. Register 2 contains the address of the class being checked at the time of the error.

System programmer response

Collect the items listed in "Diagnostics" on page 958 and contact IBM Support.

Restart the queue manager.

00C80080

A severe error has occurred during a SAF RACROUTE REQUEST=STAT call to the External Security Manager (ESM) during security switch processing at queue manager initialization time.

System action

The queue manager is terminated, and a dump is produced. Register 2 contains the address of the class being checked at the time of the error.

System programmer response

See your ESM documentation for information about any return codes that appear in the job log. Check your security configuration (for example, that the required classes are installed and active). If you are unable to resolve the problem, collect the items listed in "Diagnostics" on page 958 and contact IBM Support.

00C80081

A severe error has occurred during a SAF RACROUTE REQUEST=EXTRACT call to the External Security Manager (ESM) during security switch processing at queue manager initialization time.

System action

The queue manager is terminated, and a dump is produced. Register 2 contains the address of the entity being checked at the time of the error.

System programmer response

See your ESM documentation for information about any return codes that appear in the job log. For information about setting IBM MQ security switches, see Switch profiles. If you are unable to resolve the problem, collect the items listed in "Diagnostics" on page 958 and contact IBM Support.

00C80082

A severe error has occurred during a SAF RACROUTE REQUEST=LIST (create) call to the External Security Manager (ESM) during security switch processing at queue manager initialization time.

System action

The queue manager is terminated, and a dump is produced. Register 2 contains the address of the class, and register 3 the address of the entity, being checked at the time of the error.

System programmer response

See your ESM documentation for information about any return codes that appear in the job log. If you are unable to resolve the problem, collect the items listed in "Diagnostics" on page 958 and contact IBM Support.

00C80083

An unexpected return code has been received from one of the following SAF calls to the External Security Manager (ESM) during security switch processing at queue manager initialization time.

- RACROUTE REQUEST=EXTRACT
- RACROUTE REQUEST=LIST
- RACROUTE REQUEST=STAT

System action

Message CSQH004I is produced containing the return codes from SAF and the ESM. The queue manager is terminated, and a dump is produced. Register 2 contains the address of the return codes.

System programmer response

See your ESM documentation for information about the return codes that appear in message CSQH004I (in the job log) or the dump. For information about setting IBM MQ security switches, see Switch profiles. If you are unable to resolve the problem, collect the items listed in "Diagnostics" on page 958 and contact IBM Support.

00C80084

An unexpected setting for the subsystem security switch was encountered.

System action

The queue manager is terminated, and a dump is produced. Register 2 contains the address of the control block containing the switch setting.

System programmer response

Collect the items listed in "Diagnostics" on page 958, together with a note of what you expected the switch to be set to, and whether you had defined a profile for it or not, and contact IBM Support.

00C80090

A severe error has occurred during a SAF RACROUTE REQUEST=STAT call to the External Security Manager (ESM) during security switch processing at queue manager initialization time.

System action

The queue manager is terminated, and a dump is produced. Register 2 contains the address of the class being checked at the time of the error.

System programmer response

See your ESM documentation for information about any return codes that appear in the job log. Check your security configuration (for example, that the required classes are installed and active). If you are unable to resolve the problem, collect the items listed in "Diagnostics" on page 958 and contact IBM Support.

00C80091

A severe error has occurred during a SAF RACROUTE REQUEST=EXTRACT call to the External Security Manager (ESM) during security switch processing at queue manager initialization time.

System action

The queue manager is terminated, and a dump is produced. Register 2 contains the address of the entity being checked at the time of the error.

System programmer response

See your ESM documentation for information about any return codes that appear in the job log. For information about setting IBM MQ security switches, see Switch profiles. If you are unable to resolve the problem, collect the items listed in "Diagnostics" on page 958 and contact IBM Support.

00C80092

A severe error has occurred during a SAF RACROUTE REQUEST=LIST (create) call to the External Security Manager (ESM) during security switch processing at queue manager initialization time.

System action

The queue manager is terminated, and a dump is produced. Register 2 contains the address of the class, and register 3 the address of the entity, being checked at the time of the error.

System programmer response

See your ESM documentation for information about any return codes that appear in the job log. If you are unable to resolve the problem, collect the items listed in "Diagnostics" on page 958 and contact IBM Support.

00C80093

An unexpected return code has been received from one of the following SAF calls to the External Security Manager (ESM) during security switch processing at queue manager initialization time.

- RACROUTE REQUEST=EXTRACT
- RACROUTE REQUEST=LIST
- RACROUTE REQUEST=STAT

System action

Message CSQH004I is produced containing the return codes from SAF and the ESM. The queue manager is terminated, and a dump is produced. Register 2 contains the address of the return codes.

System programmer response

See your ESM documentation for information about the return codes that appear in message CSQH004I (in the job log) or the dump. For information about setting IBM MQ security switches, see Switch profiles. If you are unable to resolve the problem, collect the items listed in "Diagnostics" on page 958 and contact IBM Support.

00C80094

An unexpected setting for the subsystem security switch was encountered.

System action

The queue manager is terminated, and a dump is produced. Register 2 contains the address of the control block containing the switch setting.

System programmer response

Collect the items listed in "Diagnostics" on page 958, together with a note of what you expected the switch to be set to, and whether you had defined a profile for it or not, and contact IBM Support.

Restart the queue manager.

00C80095

An internal loop count was exceeded during security switch processing at queue manager initialization time.

System action

The queue manager is terminated, and a dump is produced. Register 2 contains the address of the class being checked at the time of the error.

System programmer response

Collect the items listed in “Diagnostics” on page 958 and contact IBM Support.

Restart the queue manager.

00C80100

A severe error has occurred during a SAF RACROUTE REQUEST=STAT call to the External Security Manager (ESM) during security switch processing at queue manager initialization time.

System action

The queue manager is terminated, and a dump is produced. Register 2 contains the address of the class being checked at the time of the error.

System programmer response

See your ESM documentation for information about any return codes that appear in the job log. Check your security configuration (for example, that the required classes are installed and active). If you are unable to resolve the problem, collect the items listed in “Diagnostics” on page 958 and contact IBM Support.

00C80101

A severe error has occurred during a SAF RACROUTE REQUEST=EXTRACT call to the External Security Manager (ESM) during security switch processing at queue manager initialization time.

System action

The queue manager is terminated, and a dump is produced. Register 2 contains the address of the entity being checked at the time of the error.

System programmer response

See your ESM documentation for information about any return codes that appear in the job log. For information about setting IBM MQ security switches, see Switch profiles. If you are unable to resolve the problem, collect the items listed in “Diagnostics” on page 958 and contact IBM Support.

00C80102

A severe error has occurred during a SAF RACROUTE REQUEST=LIST (create) call to the External Security Manager (ESM) during security switch processing at queue manager initialization time.

System action

The queue manager is terminated, and a dump is produced. Register 2 contains the address of the class, and register 3 the address of the entity, being checked at the time of the error.

System programmer response

See your ESM documentation for information about any return codes that appear in the job log. If you are unable to resolve the problem, collect the items listed in “Diagnostics” on page 958 and contact IBM Support.

00C80103

An unexpected return code has been received from one of the following SAF calls to the External Security Manager (ESM) during security switch processing at queue manager initialization time.

- RACROUTE REQUEST=EXTRACT
- RACROUTE REQUEST=LIST
- RACROUTE REQUEST=STAT

System action

Message CSQH004I is produced containing the return codes from SAF and the ESM. The queue manager is terminated, and a dump is produced. Register 2 contains the address of the return codes.

System programmer response

See your ESM documentation for information about the return codes that appear in message CSQH004I (in the job log) or the dump. For information about setting IBM MQ security switches, see Switch profiles. If you are unable to resolve the problem, collect the items listed in "Diagnostics" on page 958 and contact IBM Support.

00C80104

An unexpected setting for the subsystem security switch was encountered.

System action

The queue manager is terminated, and a dump is produced. Register 2 contains the address of the control block containing the switch setting.

System programmer response

Collect the items listed in "Diagnostics" on page 958, together with a note of what you expected the switch to be set to, and whether you had defined a profile for it or not, and contact IBM Support.

Restart the queue manager.

00C80105

An internal loop count was exceeded during security switch processing at queue manager initialization time.

System action

The queue manager is terminated, and a dump is produced. Register 2 contains the address of the class being checked at the time of the error.

System programmer response

Collect the items listed in "Diagnostics" on page 958 and contact IBM Support.

Restart the queue manager.

00C80200

A severe error has occurred during a SAF RACROUTE REQUEST=STAT call to the External Security Manager (ESM) during security switch processing at queue manager initialization time.

System action

The queue manager is terminated, and a dump is produced. Register 2 contains the address of the class being checked at the time of the error.

System programmer response

See your ESM documentation for information about any return codes that appear in the job log. Check your security configuration (for example, that the required classes are installed and active). If you are unable to resolve the problem, collect the items listed in "Diagnostics" on page 958 and contact IBM Support.

00C80201

A severe error has occurred during a SAF RACROUTE REQUEST=EXTRACT call to the External Security Manager (ESM) during security switch processing at queue manager initialization time.

System action

The queue manager is terminated, and a dump is produced. Register 2 contains the address of the entity being checked at the time of the error.

System programmer response

See your ESM documentation for information about any return codes that appear in the job log. For information about setting IBM MQ security switches, see Switch profiles. If you are unable to resolve the problem, collect the items listed in "Diagnostics" on page 958 and contact IBM Support.

00C80202

A severe error has occurred during a SAF RACROUTE REQUEST=LIST (create) call to the External Security Manager (ESM) during security switch processing at queue manager initialization time.

System action

The queue manager is terminated, and a dump is produced. Register 2 contains the address of the class, and register 3 the address of the entity, being checked at the time of the error.

System programmer response

See your ESM documentation for information about any return codes that appear in the job log. If you are unable to resolve the problem, collect the items listed in "Diagnostics" on page 958 and contact IBM Support.

00C80203

An unexpected return code has been received from one of the following SAF calls to the External Security Manager (ESM) during security switch processing at queue manager initialization time.

- RACROUTE REQUEST=EXTRACT
- RACROUTE REQUEST=LIST
- RACROUTE REQUEST=STAT

System action

Message CSQH004I is produced containing the return codes from SAF and the ESM. The queue manager is terminated, and a dump is produced. Register 2 contains the address of the return codes.

System programmer response

See your ESM documentation for information about the return codes that appear in message CSQH004I (in the job log) or the dump. For information about setting MQ security switches, see Switch profiles. If you are unable to resolve the problem, collect the items listed in "Diagnostics" on page 958 and contact IBM Support.

00C80204

An unexpected setting for the subsystem security switch was encountered.

System action

The queue manager is terminated, and a dump is produced. Register 2 contains the address of the control block containing the switch setting.

System programmer response

Collect the items listed in "Diagnostics" on page 958, together with a note of what you expected the switch to be set to, and whether you had defined a profile for it or not, and contact IBM Support.

Restart the queue manager.

00C80205

An internal loop count was exceeded during security switch processing at queue manager initialization time.

System action

The queue manager is terminated, and a dump is produced. Register 2 contains the address of the class being checked at the time of the error.

System programmer response

Collect the items listed in "Diagnostics" on page 958 and contact IBM Support.

Restart the queue manager.

00C80206

An unexpected setting for the request type was encountered on an authentication request.

System action

The current execution unit terminates with a completion code of X'5C6' and a dump is produced. Register 2 contains the request type in error.

System programmer response

Collect the items listed in “Diagnostics” on page 958 and contact IBM Support.

Restart the queue manager.

00C80207

An unexpected setting for the request type was encountered on an authentication request.

System action

The queue manager terminates and a dump is produced. Register 2 contains the request type in error.

System programmer response

Collect the items listed in “Diagnostics” on page 958 and contact IBM Support.

Restart the queue manager.

00C81000

A severe error has occurred while processing a REFRESH SECURITY command.

System action

The current execution unit terminates with error reason code X'5C6', and a dump is produced. Register 2 contains the address of the control block involved in the error.

System programmer response

Collect the items listed in “Diagnostics” on page 958 and contact IBM Support.

z/OS *Data manager codes (X'C9')*

If a data manager reason code occurs that is not listed here, an internal error has occurred. Collect the following diagnostic items and contact IBM Support.

Diagnostics

- A description of the actions that led to the error or, if applicable, either a listing of the application program or the input string to a utility program that was being run at the time of the error.
- The console output for the period leading up to the error.
- The queue manager job log.
- The system dump resulting from the error.
- If you are using CICS, the CICS transaction dump output.
- Appropriate IBM MQ, z/OS, Db2, CICS, and IMS service levels.
- If you are using the IBM MQ Operations and Control panels, the ISPF panel name.

00C90100

The object IBM MQ was trying to create was too large to be stored.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 978 and contact IBM Support.

00C90200

A page set page retrieved was not valid.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 978 and contact IBM Support.

00C90201

A page set page retrieved was not valid. The page was not a header page.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 978 and contact IBM Support.

00C90202

A page set page retrieved was not valid. The page was not a data page.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 978 and contact IBM Support.

00C90300

MQ was unable to start a unit of recovery for this execution unit.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 978 and contact IBM Support.

00C90301

An internal logging error has occurred for the current execution unit.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 978 and contact IBM Support.

00C90400

The data manager has detected in invalid log record.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 978 and contact IBM Support.

00C90401

The data manager has detected an invalid log record subtype.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 978 and contact IBM Support.

00C90500

The data manager was asked to make a change to some data in a page, but the change would have extended beyond the specific data item.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 978 and contact IBM Support.

00C90600

The data manager was unable to locate a specific logical record within a data page. The record was required for an update, or to insert a new record immediately after.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 978 and contact IBM Support.

00C90700

The data manager was unable to locate its *resource access list entry* (RALE).

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 978 and contact IBM Support.

00C90800

The data manager was requested to put a message on a queue, but told to give the message an invalid priority.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 978 and contact IBM Support.

00C90900

The data manager was asked to retrieve a logical record from a page, but on retrieving it discovered that the record is invalid.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 978 and contact IBM Support.

00C90A00

The data manager was asked to carry out a value logging operation with an invalid length field.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 978 and contact IBM Support.

00C90B00

The space reclamation routines have been asked to deallocate a page that is not in a state to be deallocated.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 978 and contact IBM Support.

00C90C00

An object type description passed to the data manager is not valid.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 978 and contact IBM Support.

00C90D00

A page set that was originally page set n is being presented as being a different page set, probably because the started task JCL procedure for the queue manager has been changed. Register 0 contains the identifier of the page set in error, and register 2 contains the identifier it was previously associated with.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Check the started task JCL procedure for the queue manager, and undo any changes to the CSQPnnnn DD statements that specify the page sets. Restart the queue manager. If the problem persists, or no changes have been made to these statements, collect the items listed in “Diagnostics” on page 978 and contact IBM Support.

00C90D01

Your data set is not recognized as an IBM MQ page set. This may be for one of the following reasons.

- The data set has not been formatted
- You are attempting to backwards migrate from a newer version of IBM MQ to an older version of IBM MQ without first running START QMGR BACKMIG(*target-vrm*) at the newer version of IBM MQ.

Register 0 contains the identifier of the page set in error.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Investigate the reason code and take one of the following actions:

- Format the page set
- Run `START QMGR BACKMIG(`*`target-vrm`*`)` at the newer version of IBM MQ before performing the backwards migration
- Start the queue manager with the correct level of code

00C90D02

This reason code is caused by one of the following:

- An attempt to use a page set that is a valid IBM MQ page set, but does not belong to this queue manager
- An attempt to change the subsystem name

Neither of these actions is allowed.

Register 0 contains the identifier of the page set in error.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

If you were attempting to use a page set from another queue manager, correct the error. Do not attempt to change the name of your queue manager.

00C90D03

An internal error has occurred during processing of an MQGET call with the Mark Skip Backout option.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 978 and contact IBM Support.

00C90D04

During restart, the queue manager detected that a page set has been truncated. This is probably because the data set allocated during restoration of a backup was smaller than required to hold the backed up data, and so the data has been truncated. It might also occur if page set 0 is larger than the maximum supported page set size.

System action

The identifier of the page set in error is put in register 0. Restart is terminated.

System programmer response

Reallocate the data set correctly, restore the backed up data if necessary, and restart the queue manager.

00C90E00

The data manager was passed an invalid parameter describing the location of a logical record within a data page and page set.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 978 and contact IBM Support.

00C90F00

The data manager was requested to update a logical record within a page, but the record had been deleted previously.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in "Diagnostics" on page 978 and contact IBM Support.

00C90F03

A chain loop was detected during queue index rebuild processing.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in "Diagnostics" on page 978 and contact IBM Support.

00C91000

The data manager was asked to retrieve a message from an object that was not a local queue.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in "Diagnostics" on page 978 and contact IBM Support.

00C91094, 00C91095, 00C91096, 00C91097

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'. In some cases, the queue manager might terminate with completion code X'6C6'.

System programmer response

Collect the items listed in "Diagnostics" on page 978 and contact IBM Support.

00C91101

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in "Diagnostics" on page 978 and contact IBM Support.

00C91102

MQ received a return code indicating an error from the RRS ATRSROI service.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

The return code from ATRSROI is in register 15. See the *z/OS MVS Programming: Resource Recovery* manual for information about the return code.

00C91104

The data manager was requested to carry out a browse message operation, but the required lock was not held.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in "Diagnostics" on page 978 and contact IBM Support.

00C91200

The internal data manager locate-object routine could not find the object it was seeking during UNDO processing.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in "Diagnostics" on page 978 and contact IBM Support.

00C91300

During queue manager startup, an attempt was made to recover an object, the length of which exceeds a single data page. However, one of the intermediate data pages was not available, and IBM MQ was unable to recover the object.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in "Diagnostics" on page 978 and contact IBM Support.

00C91400

The data manager was unable to access the header page (page 0) of one of the page sets.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced. The number of the page set with a header page that was unreadable is held in register 2.

System programmer response

1. Check for a preceding IEC161I, CSQP060E, or CSQP011E message relating to page set mentioned in register 2.
2. Check the following:
   - For the page set mentioned in register 2, is the appropriate CSQPnnnn DD statement present in the started task JCL procedure for the queue manager, xxxxMSTR?
   - Does this DD statement reference a genuine data set? DD DUMMY statements are not allowed for page sets.
   - Is DEFINE PSID(nn) present in the CSQINP1 initialization input data set?
3. If you are still unable to resolve the problem, collect the items listed in "Diagnostics" on page 978 and contact IBM Support.

00C91500

During queue manager startup, the data manager was following a chain of objects on disk, and requested the next data page in the chain from the buffer manager. However, the buffer manager could not supply this page.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 978 and contact IBM Support.

00C91600

During restart, the data manager rebuilds its in-storage structures from page set data. On rebuilding an object, data manager discovered that the object already exists.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 978 and contact IBM Support.

00C91700, 00C91800

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 978 and contact IBM Support.

00C91900

During restart, data manager has detected an error in the rebuild of its in-storage object structures.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 978 and contact IBM Support.

00C91B01

During restart, the data manager found a queue with messages that are apparently located in a newly added page set. This is probably because the queue manager was run with a page set offline, and a new page set was formatted to replace the original one. This will lead to data loss.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 978 and contact IBM Support.

00C91C00

A delete purge request has been issued but the object type is not a local queue.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 978 and contact IBM Support.

00C91D00

A lock request has failed during an attempt to lock all pages associated with a long catalog object, or a long message.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 978 and contact IBM Support.

00C91E00

During a request issued by CSQIPUT5 or CSQIPUT6, an attempt to obtain a page level lock was unsuccessful.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 978 and contact IBM Support.

00C91F00

During a request issued by CSQIPUT5 or CSQIPUT6, an attempt to obtain a record level lock was unsuccessful.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 978 and contact IBM Support.

00C92000

An attempt to obtain a page level lock on the owner page relating to an object or message was unsuccessful.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 978 and contact IBM Support.

00C92100

An attempt to obtain a page level lock while trying to insert data was unsuccessful.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 978 and contact IBM Support.

00C92200

An attempt to obtain a record level lock while trying to insert data was unsuccessful.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 978 and contact IBM Support.

00C92300

An attempt to obtain a record level lock while trying to amend data was unsuccessful.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 978 and contact IBM Support.

00C92400

An attempt to get a lock on object type concatenated with object name within CSQIMGE1 was unsuccessful.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 978 and contact IBM Support.

00C92500, 00C92600, 00C92700, 00C92800, 00C92900, 00C92A00, 00C92B00, 00C92C00, 00C92D00, 00C92E00, 00C92F00, 00C93000

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 978 and contact IBM Support.

00C93100

A keyed read queue has encountered an error. A problem has occurred in the hash-table structure for the queue.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 978 and contact IBM Support.

00C93200, 00C93300

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 978 and contact IBM Support.

00C93500

IBM MQ was extending a page set at startup, based on log records from earlier dynamic page set extend operations. (IBM MQ does this so that any media recovery operation will have the required number of pages available in the page set.)

The page set could not be extended to the required RBA value.

The contents of the relevant registers are as follows:

R0
The number of the page set that could no longer be extended

R2
The logged page number that IBM MQ was trying to extend to

R3
The high page number at restart. This is the base from which IBM MQ was extending.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Create a larger page set, using multiple disk volumes if required, with a larger secondary extent value. The high page number of the page set should at least match that shown in register 2 in the dump.

00C93700

A queue contains messages, but the storage class named in the queue definition does not exist. This is an error.

This reason code is issued on queue manager restart if it is **not** the first time the queue manager has been started after migration to a new version.

Register 2 contains the first 4 characters of the storage class name, and register 3 contains characters 5 through 8.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the dump and a listing of your page set 0 and contact IBM Support.

00C93800

A queue contains messages, which are on a page set other than that defined by the storage class named by the queue.

This reason code is issued on queue manager restart if it is **not** the first time the queue manager has been started after migration to a new version. It is preceded by one or more instances of message CSQI028E.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the dump and a listing of your page set 0 and contact IBM Support.

00C93900

During MQPUT processing, IBM MQ was unable to acquire a lock on the storage class of the queue.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 978 and contact IBM Support.

00C93A00

During MQGET processing, IBM MQ was unable to acquire a lock on the queue it was processing.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 978 and contact IBM Support.

00C93B00

During MQPUT processing, IBM MQ was unable to acquire a lock on the queue it was processing.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in "Diagnostics" on page 978 and contact IBM Support.

00C93C00

During MQGET processing, IBM MQ was unable to retrieve a message page from a queue it was processing.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in "Diagnostics" on page 978 and contact IBM Support.

00C93D00, 00C93E00, 00C93F00, 00C94000, 00C94100

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in "Diagnostics" on page 978 and contact IBM Support.

00C94200

MQ received a return code indicating an error from the RRS ATREINT service.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

The return code from ATREINT is in register 15. See the *z/OS MVS Programming: Resource Recovery* manual for information about the return code.

00C94201

MQ received a return code indicating that RRS is not available from the RRS ATREINT service. This can occur if RRS is stopped when running an IBM MQ application linked with an RRS stub.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Ensure RRS is running when using an IBM MQ application linked with an RRS stub

00C94300

MQ received a return code indicating an error from the RRS ATRSIT service.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

The return code from ATRSIT is in register 15. See the *z/OS MVS Programming: Resource Recovery* manual for information about the return code.

00C94400

MQ received a return code indicating an error from the RRS ATRSPID service.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

The return code from ATRSPID is in register 15. See the *z/OS MVS Programming: Resource Recovery* manual for information about the return code.

00C94500, 00C94501, 00C94502

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in "Diagnostics" on page 978 and contact IBM Support.

00C94503

A page set that has been the subject of the RESETPAGE function had not previously been through a clean shutdown of the queue manager. Using this page set for subsequent IBM MQ processing would lead to data integrity problems.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Check the page sets that are defined to the queue manager. One or more of the page sets has been the subject of a RESETPAGE operation. Do not run the RESETPAGE operation against page sets that are either of the following:

- Fuzzy page set backups
- From a queue manager that has terminated abnormally

If you are unable to solve the problem, collect the items listed in "Diagnostics" on page 978 and contact IBM Support.

00C94505

An internal error has occurred.

An attempt to restart with a log from another queue manager was detected. The queue sharing group name recorded in the log during checkpoint does not match the name of the queue sharing group in the queue manager using that log. If the correct log is being used, you can perform the change only after a clean shutdown of the queue manager, that is, after a quiesce.

Message CSQI060E is issued before this error occurs.

System action

Restart is terminated abnormally with completion code X'5C6' and a dump is produced.

System programmer response

Restart the queue manager using the correct logs and BSDS, or change the QSGDATA system parameter. Note that you cannot change the name of the queue sharing group that a queue manager uses unless it has been shut down normally.

The following registers in the dump contain helpful values:

- R0 = the queue sharing group name recorded in the log
- R2 = the queue sharing group name in the running queue manager

00C94506

An internal error has occurred.

An attempt to restart with a log from another queue manager was detected. The shared queue manager identifier recorded in the log during checkpoint does not match the shared queue manager identifier in the queue manager using that log. If the correct log is being used, the entry in the Db2 CSQ.ADMIN_B_QMGR table for this queue manager has been corrupted.

Message CSQI061E is issued before this error occurs.

System action

Restart is terminated abnormally with completion code X'5C6' and a dump is produced.

System programmer response

Restart the queue manager using the correct logs and BSDS. If the correct logs are being used, correct the entry for the queue manager in the Db2 CSQ.ADMIN_B_QMGR table. If you cannot resolve the problem, contact IBM Support for assistance.

The following registers in the dump contain helpful values:

- R0 = the queue manager identifier recorded in the log
- R2 = the queue manager identifier in the running queue manager

00C94507

An internal error has occurred during processing of Mark Skip Backout.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 978 and contact IBM Support.

00C94510

A request was made to a coupling facility resource manager service within IBM MQ. The coupling facility resource manager service returned an unexpected return code.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 978 and contact IBM Support.

00C94511

An attempt to obtain storage for the data manager's use was unsuccessful. This indicates there could be a wider-ranging problem relating to storage availability.

System action

The queue manager is terminated and a dump is produced.

System programmer response

Check that you are running with the recommended region size, and if not, reset your system and restart the queue manager. If this is not the cause, use these items to diagnose the cause of the problem:

- Queue manager job log
- Information about any other storage-related problems
- System dump resulting from the error

00C94512

A request was made to a Db2 resource manager service within IBM MQ. The Db2 resource manager service returned an unexpected return code.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 978 and contact IBM Support.

00C94513

A request was made to a coupling facility resource manager service within IBM MQ. The coupling facility resource manager service returned an unexpected return code.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 978 and contact IBM Support.

00C9451A

A request was made to a Db2 resource manager service within IBM MQ during restart. The Db2 resource manager service returned an unexpected return code related to a locked table condition.

System action

The queue manager terminates with completion code X'5C6', and a dump is produced.

System programmer response

Restart the queue manager. If you started several queue managers at the same time, try restarting them one at a time to alleviate this condition.

If the problem persists, collect the items listed in “Diagnostics” on page 978 and contact IBM Support.

00C94525

An internal error has occurred.

An attempt to restart with a page set that has not had RESETPAGE run was detected.

Message CSQI050E is issued before this error occurs.

System action

Restart is terminated abnormally with completion code X'5C6' and a dump is produced.

System programmer response

Restart the queue manager, after running CSQUTIL, with the RESETPAGE option for the page set or sets indicated in the CSQI050E messages

00C94531

An internal error has occurred.

An attempt to restart whilst page set 0 has not had RESETPAGE run was detected.

Message CSQI053E is issued before this error occurs and will indicate page set 0 is the affected page set.

System action

Restart is terminated abnormally with completion code X'5C6' and a dump is produced.

System programmer response

Contact IBM Support.

00C94526

During MQOPEN processing, IBM MQ detected the presence of duplicate queue control blocks.

System action

The MQOPEN call fails with reason code MQRC_OBJECT_NOT_UNIQUE and a dump is produced.

System programmer response

Collect the items listed in "Diagnostics" on page 978 and contact IBM Support.

00C9FEEE

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in "Diagnostics" on page 978 and contact IBM Support.

Related tasks

Collecting troubleshooting information

z/OS *Recovery log manager codes (X'D1')*

If a recovery log manager reason code occurs that is not listed here, an internal error has occurred. Collect the following diagnostic items and contact IBM Support.

Diagnostics

- The console output for the period leading up to the error.
- The system dump resulting from the error.
- If you are using CICS , the CICS transaction dump output.
- Appropriate IBM MQ, z/OS, Db2, CICS, and IMS service levels.
- A printout of SYS1.LOGREC, if the reason code is issued by an active queue manager.
- If the reason code is issued by an active queue manager, a CSQ1LOGP detail report containing the log records associated with the problem.
- Contents of the BSDS. Obtain a listing by running the Print Log Map utility (CSQJU004).
- The recovery log manager standard diagnostic information, which is provided in the SYS1.LOGREC variable recording area (VRA) of the system diagnostic work area (SDWA) for many of the reason codes:

  MODID
  Name of the module issuing the error.

  LEVEL
  Change level.

  COMPONENT
  Subcomponent identifier of the recovery log manager.

  REGISTERS
  General purpose registers (GPRs) 0-15 at time of the abend.

00D10010

The end log range value specified on an invocation of the log print utility (CSQ1LOGP) is less than or equal to the start range value.

System action

No error is issued by log services, and no information is written to SYS1.LOGREC data set.

System programmer response

Correct the log range input control parameters specified in the invocation of the log print utility.

For more information about log services, refer to CSQ1LOGP.

00D10011

An invocation of the log print utility (CSQ1LOGP) was unable to obtain the storage required to perform the request.

System action

No error is issued by log services, and no information is written to SYS1.LOGREC data set.

System programmer response

It is probable that the REGION parameter on the EXEC statement of the job control language (JCL) for this invocation is too small. Increase the REGION size, and resubmit the log print request.

For more information about log services, refer to Address space storage.

00D10012

An invocation of the log print utility (CSQ1LOGP) was unsuccessful because the job control language (JCL) for this invocation did not specify either the use of the bootstrap data set (BSDS) or, in the absence of the BSDS, the active or archive log data sets.

System action

No error is issued by log services, and no information is written to SYS1.LOGREC data set.

System programmer response

Correct the JCL and resubmit the log print request.

For more information about BSDS, refer to Managing the BSDS.

00D10013

An invocation of the log print utility (CSQ1LOGP) resulted in a VSAM error while attempting to open the bootstrap data set (BSDS).

This reason code, and the VSAM return code are issued with message CSQ1221E.

System action

No error is issued by log services, and no information is written to SYS1.LOGREC data set.

System programmer response

Refer to *z/OS DFSMS Macro Instructions for Data Sets* to determine the meaning of the VSAM OPEN error. Take appropriate action, and resubmit the log print request.

00D10014

The job control language (JCL) for an invocation of the log print utility (CSQ1LOGP) specified the use of the bootstrap data set (BSDS), but the utility control statements did not specify values for RBASTART and RBAEND.

The RBASTART and RBAEND values must be specified when using the BSDS, although they are not required when using the active or archive logs.

System action

No error is issued by log services, and no information is written to SYS1.LOGREC data set.

System programmer response

Either:

- Continue to use the BSDS, but change the utility control statements to specify values for RBASTART and RBAEND
- Change the JCL to use the active and archive data sets instead

For more information, refer to CSQ1LOGP.

00D10015

An invocation of the log print utility (CSQ1LOGP) was unsuccessful because the record format of the bootstrap data set is incompatible with this release of the log print services.

System action

No error is issued by log services, and no information is written to SYS1.LOGREC data set.

System programmer response

Ensure that the correct release of the log print services are used with the appropriate BSDS record format.

For more information, refer to CSQ1LOGP.

00D10019

An invocation of the log print utility (CSQ1LOGP) resulted in a VSAM error while attempting to open the bootstrap data set (BSDS). The error was determined to be one which could be corrected by use of a VSAM access method services (AMS) VERIFY call, but the VERIFY call was also unsuccessful.

System action

No error is issued by log services, and no information is written to SYS1.LOGREC data set.

System programmer response

Collect the following items, and contact IBM Support:

- A copy of the user's job control language (JCL) that was used to invoke the log print utility (CSQ1LOGP)
- The log data sets that the user was attempting to print

00D10020

The log print utility (CSQ1LOGP) issued this message because the end of data has been reached (that is, the end of the log, or the end of the user-specified data sets, or the user-specified RBAEND value has been reached).

System action

No error is issued by log services, and no information is written to SYS1.LOGREC data set.

System programmer response

This is not an error. This reason code denotes a normal end of data condition. No action is necessary.

For more information, refer to CSQ1LOGP.

00D10021

An invocation of the log print utility (CSQ1LOGP) encountered a gap in the log RBA range when switching log data sets. This indicates that log records might be missing.

Normally, a continuous set of log records is supplied as input by the ACTIVE and ARCHIVE DDnames (or the BSDS DDname if you are using the bootstrap data set (BSDS) to access the log data sets) in the job control language (JCL) used to invoke the utility. If a log data set was removed from the JCL, this condition will arise.

System action

No error is issued by log services, and no information is written to SYS1.LOGREC data set.

System programmer response

If the log data set was not removed intentionally, check the JCL to ensure that the log data sets are specified in ascending RBA value order. If you are using the BSDS to access the log data sets, use the print log map utility (CSQJU004) to examine the RBA ranges as recorded in the BSDS, and note any RBA gaps that might have resulted from the deletion of an active or archive log data set.

If it appears that a log error might have occurred, see Active log problems for information about dealing with problems on the log.

00D10022

An invocation of the log print utility (CSQ1LOGP) encountered a gap in the log RBA range when switching log data sets. This indicates that log records might be missing. The log RBA of the next record following the gap is greater than the RBAEND value specified in the utility control statements.

Normally, a continuous set of log records is supplied as input by the ACTIVE and ARCHIVE DDnames (or the BSDS DDname if using the bootstrap data set (BSDS) to access the log data sets) in the job control language (JCL) used to invoke the utility. If a log data set was removed from the JCL, this condition will arise.

System action

No error is issued by log services, and no information is written to SYS1.LOGREC data set.

System programmer response

Check the JCL and the RBAEND value specified in the utility control statements.

If a log data set was not removed intentionally, check that the log data sets are specified in ascending RBA value order. If using the BSDS to access log data sets, use the print log map utility (CSQJU004) to examine the RBA ranges as recorded in the BSDS, and note any RBA gaps that might have resulted from the deletion of an active or archive log data set.

If it appears that a log error might have occurred, see Active log problems for information about dealing with problems on the log.

00D10024

An invocation of the log print utility (CSQ1LOGP) encountered a log RBA sequence error. The RBA of the previous log record is greater than the RBA of the current log record.

Normally, a continuous set of log records is supplied as input by the ACTIVE and ARCHIVE DDnames (or the BSDS DDname if using the bootstrap data set (BSDS) to access the log data sets) in the job control language (JCL) used to invoke the utility. If a log data set appears out of sequence, this condition will arise.

System action

No error is issued by log services, and no information is written to SYS1.LOGREC data set.

System programmer response

Check the JCL to ensure that the log data sets are specified in ascending RBA value order. If using the BSDS to access the log data sets, use the print log map utility (CSQJU004) to examine the RBA ranges associated with each archive and active log data set. If both archive and active log data sets are used, the first archive log data set must contain the lowest log RBA value. If necessary, adjust the concatenation of the archive and active log data sets in the JCL to ensure that log records are read in ascending RBA sequence, and resubmit the log print request.

If it appears that a log error might have occurred, see Active log problems for information about dealing with problems on the log.

00D10025

An invocation of the log print utility (CSQ1LOGP) resulted in a VSAM GET error while attempting to read the active log data set.

This reason code, and the VSAM return and reason codes are issued in message CSQ1221E.

System action

No error is issued by log services, and no information is written to SYS1.LOGREC data set.

System programmer response

Refer to *z/OS DFSMS Macro Instructions for Data Sets* to determine the meaning of the VSAM GET error and the RPL error code. Take appropriate action to correct the error, and resubmit the log print request.

00D10026

An invocation of the log print utility (CSQ1LOGP) was unsuccessful because an RBA value within the range specified by RBASTART and RBAEND could not be located on a log data set.

This reason code, and the RBA value that could not be located are issued with message CSQ1216E

System action

No error is issued by log services, and no information is written to SYS1.LOGREC data set.

System programmer response

Check the utility control statements to ensure that the RBASTART and RBAEND values have not exceeded the lower or upper bounds of the RBAs available on all the active or archive log data sets defined by DDnames in the JCL.

If you are using the BSDS to access the log data sets, use the print log map utility (CSQJU004) to examine the RBA ranges associated with each archive and active log data set.

Correct the JCL and utility control statements as necessary, and resubmit the log print request.

For more information, refer to CSQ1LOGP.

00D10027

An invocation of the log print utility (CSQ1LOGP) resulted in a VSAM GET error while attempting to read the bootstrap data set (BSDS).

This reason code, and the VSAM return and reason codes, are issued with message CSQ1221E.

System action

No error is issued by log services, and no information is written to SYS1.LOGREC data set.

System programmer response

Refer to *z/OS DFSMS Macro Instructions for Data Sets* to determine the meaning of the VSAM GET error and the RPL error code. Take appropriate action to correct the error and resubmit the log print request.

00D1002A

An invocation of the log print utility (CSQ1LOGP) was unsuccessful because an RBA value has been requested in an active log data set that has previously not been opened. A VSAM OPEN error occurred while attempting to open the active log data set.

This reason code, and the VSAM return and reason codes, are issued in message CSQ1221E.

System action

No error is issued by log services, and no information is written to SYS1.LOGREC data set.

System programmer response

Refer to *z/OS DFSMS Macro Instructions for Data Sets* to determine the meaning of the VSAM OPEN error and the ACB error code. Take appropriate action to correct the error, and resubmit the log print request.

00D1002B

An invocation of the log print utility (CSQ1LOGP) was unsuccessful because an RBA value has been requested in an active log data set that has previously not been opened. A VSAM OPEN error occurred while attempting to open the active log data set. The VSAM OPEN error was determined to be one that could be corrected, however, a system error occurred while executing a z/OS TESTCB macro to determine whether the active log data set in question was a VSAM ESDS (entry-sequenced data set) or a VSAM LDS (linear data set).

This reason code, and the VSAM return and reason codes are issued in message CSQ1221E.

System action

No error is issued by log services, and no information is written to SYS1.LOGREC data set.

System programmer response

Refer to *z/OS DFSMS Macro Instructions for Data Sets* to determine the meaning of the VSAM OPEN error and the ACB error code. Take appropriate action to correct the error, and resubmit the log print request.

If the problem persists, collect the following items, and contact IBM Support:

- A copy of the job control language (JCL) used to invoke the log print utility (CSQ1LOGP)
- The log data sets that the user was attempting to print

00D1002C

An invocation of the log print utility (CSQ1LOGP) was unsuccessful because an RBA value has been requested in a active log data set that has previously not been opened. A VSAM OPEN error occurred while attempting to open the active log data set. The VSAM OPEN error was determined to be one which could be corrected by use of a VSAM access method services (AMS) VERIFY call, but the VERIFY call was unsuccessful.

This reason code, and the VSAM return and reason codes are issued with message CSQ1221E.

System action

No error is issued by log services, and no information is written to SYS1.LOGREC data set.

System programmer response

Refer to *z/OS DFSMS Macro Instructions for Data Sets* to determine the meaning of the VSAM OPEN error and the ACB error code. Take appropriate action to correct the error, and resubmit the log print request.

00D1002D

An invocation of the log print utility (CSQ1LOGP) was unsuccessful because an RBA value has been requested in an active log data set that has previously not been opened. A VSAM OPEN error occurred while attempting to open the active log data set. The VSAM OPEN error was corrected by use of a VSAM access method services (AMS) VERIFY call, but a subsequent attempt to reposition the VSAM pointer back to the beginning of the active log data set (using the VSAM AMS POINT call) was unsuccessful.

This reason code and the VSAM return and reason codes are issued with message CSQ1221E.

System action

No error is issued by log services, and no information is written to SYS1.LOGREC data set.

System programmer response

Refer to *z/OS DFSMS Macro Instructions for Data Sets* to determine the meaning of the VSAM OPEN error and the ACB error code. Take the appropriate action to correct the error, and resubmit the print log request.

00D10030

An invocation of the log print utility resulted in an internal error.

System action

No error is issued by log services, and no information is written to SYS1.LOGREC data set.

System programmer response

Collect the following items, and contact IBM Support:

- A copy of the job control language (JCL) used to invoke the log print utility
- The log data sets that the user was attempting to print

00D10031

An invocation of the log print utility (CSQ1LOGP) was unsuccessful because an RBA value has been requested in a log data set that has previously not been opened. The job control language (JCL) has specified that the bootstrap data set (BSDS) be used as the guide to determine which data sets are required. An attempt to allocate the appropriate data set dynamically (using z/OS SVC 99) was unsuccessful.

This reason code, and the dynamic allocation information and error codes (S99INFO and S99ERROR) are issued with message CSQ1222E.

System action

No error is issued by log services, and no information is written to SYS1.LOGREC data set.

System programmer response

Go to the Interpreting DYNALLOC return codes topic of the *MVS Authorized Assembler Services Guide* for information about these codes. Take the appropriate action to correct the error, then resubmit the log print request.

00D10040

An invocation of the log print utility (CSQ1LOGP) was unsuccessful because an RBA value has been requested in an archive log data set (on tape) that has previously not been opened. An attempt was made to open the second file on the archive log tape (the first file normally contains the bootstrap data set) but this was unsuccessful because the archive log data set was not the second file on the archive log tape. The read job file control block (RDJFCB) macro was then invoked to attempt to change the data set sequence number from the default value of 2 to a value of 1, before attempting to open the second file again, but the macro invocation resulted in an error.

This reason code, and the RDJFCB return code are issued in message CSQ1223E.

System action

No error is issued by log services, and no information is written to SYS1.LOGREC data set.

System programmer response

Refer to the *z/OS DFSMSdfp Advanced Services* manual to determine the meaning of the RDJFCB error code. Take the appropriate action to correct the error, and resubmit the log print request.

00D10044

An invocation of the log print utility (CSQ1LOGP) was unsuccessful because an RBA value has been requested in an archive log data set that has previously not been opened. An attempt to open the archive log data set resulted in a QSAM (queued sequential access method) error.

System action

No error is issued by log services, and no information is written to SYS1.LOGREC data set.

System programmer response

Check the console for messages indicating the cause of the QSAM error. Take the appropriate action to correct the error, and resubmit the log print request.

00D10048

An invocation of the log print utility (CSQ1LOGP) was unsuccessful because a QSAM (queued sequential access method) GET error occurred while reading an archive log data set.

System action

No error is issued by log services, and no information is written to SYS1.LOGREC data set.

System programmer response

Check the console for messages indicating the cause of the QSAM error. Take the appropriate action to correct the error, and resubmit the log print request.

00D10050

An invocation of the log print utility (CSQ1LOGP) was unsuccessful because the bootstrap data set (BSDS) was erroneously specified as one of the archive data sets in the job control language (JCL).

System action

No error is issued by log services, and no information is written to SYS1.LOGREC data set.

System programmer response

Examine the JCL, and remove the occurrence of the BSDS data set as one of the concatenated ARCHIVE data sets. Resubmit the log print request.

For more information, refer to Archive log problems.

00D10061

An invocation of the log print utility (CSQ1LOGP) succeeded, but an unexpected physical record length was encountered for the log record control interval (CI) for an active or archive log data set.

The data on the log data set might have been corrupted after it was written by IBM MQ. The data in the log data set might still be usable, but with caution.

The length of a log CI in an active log data set is expected to be 4096 bytes. The length of a log CI in an archive log data set is expected to be 4089 bytes.

System action

No error is issued by log services, and no information is written to SYS1.LOGREC data set. The log print request has completed. This reason code is issued as a warning.

System programmer response

Ensure that the ACTIVE and ARCHIVE DDnames in the job control language (JCL) refer to active and archive logs correctly.

If the problem persists, collect the following items, and contact IBM Support:

- A copy of the job control language (JCL) used to invoke the log print utility (CSQ1LOGP)
- The log data set that the user was trying to print

00D10062

An invocation of the log print utility (CSQ1LOGP) succeeded, but the first log record segment could not be found for a middle spanned log record segment.

System action

No error is issued by log services, and no information is written to SYS1.LOGREC data set. The log print request has completed. This reason code is issued as a warning.

System programmer response

Several possibilities exist for the cause of this condition:

- The recovery log manager component of IBM MQ did not originally construct the log record header (LRH) properly
- The LRH for the log record segment was damaged after it was written by IBM MQ
- The application program continued to process after being informed about a gap in the log RBA values (reason code X'00D10021')

Determine if the LRH of the log record segment is truly in error by looking at the record segments directly preceding and after the record segment in question.

Take the appropriate action to correct the error, and resubmit the log print request. If the problem persists, collect the following items, and contact IBM Support:

- A copy of the job control language (JCL) used to invoke the lop print utility (CSQ1LOGP)
- The log data set that the user was attempting to print

00D10063

An invocation of the log print utility (CSQ1LOGP) succeeded, but the first log record segment could not be found for a last spanned log record segment.

System action

No error is issued by log services, and no information is written to SYS1.LOGREC data set. The log print request has completed. This reason code is issued as a warning.

System programmer response

Several possibilities exist for the cause of this condition:

- The recovery log manager component of IBM MQ did not originally construct the log record header (LRH) properly
- The LRH for the log record segment was damaged after it was written by IBM MQ
- The application program continued to process after being informed about a gap in the log RBA values (reason code X'00D10021')

Determine if the LRH of the log record segment is truly in error by looking at the record segments directly before and after the record segment in question.

Take the appropriate action to correct the error, and resubmit the log print request. If the problem persists, collect the following items, and contact IBM Support:

- A copy of the job control language (JCL) used to invoke the lop print utility (CSQ1LOGP)
- The log data set that the user was attempting to print

00D10114

IBM MQ failed to read or write member information in the queue sharing group table, CSQ.ADMIN_B_QSG.

System action

Queue manager initialization terminates.

System programmer response

Investigate Db2 SQL errors reported in the queue manager job log immediately preceding this error, to determine the cause. It is most likely due to incorrect table setup, plans not bound or insufficient authority to execute Db2 plans.

00D10121

The BSDS is not valid. A non-valid BSDS is the result of a failure during a previous attempt to run the BSDS conversion utility.

System action

Queue manager startup terminates.

System programmer response

The procedure for running the BSDS conversion utility involves renaming the original BSDS. Restore the BSDS to the original pre-conversion copy by renaming the data sets, then try the conversion again.

When the conversion is successful, try the program that issued the error message again.

00D10122

The BSDS version is not supported by this release of IBM MQ.

System action

Queue manager startup, or the process that was accessing the BSDS, terminates.

System programmer response

Start the queue manager at a version of IBM MQ that supports the BSDS version.

You can determine the version of a BSDS by running the print log map utility (CSQJU004)

00D10250

An unrecoverable error occurred while updating either the BSDS or the z/OS catalog to reflect changes in active log data sets.

System action

An execution unit writes a record to SYS1.LOGREC and requests an SVC dump. The queue manager then terminates abnormally.

System programmer response

Obtain the SYS1.LOGREC and SVC dump. Correct the error, and restart the queue manager.

You might find the items listed in "Diagnostics" on page 993 useful in resolving the problem. In addition, see the description of reason code X'00D10252' for details of the information recorded in the variable recording area (VRA) of the system diagnostic work area (SDWA).

Examine the console log for a CSQJxxxx message preceding this error to determine whether the error was a BSDS error or a z/OS catalog update error. If you cannot resolve the problem, contact Support.

00D10251

An unrecoverable error occurred in the log buffer writer.

System action

An execution unit writes a record to SYS1.LOGREC and requests an SVC dump. The queue manager then terminates abnormally.

System programmer response

Obtain the SYS1.LOGREC and the SVC dump. This error is usually caused by a previous error that was recorded on SYS1.LOGREC and produced an SVC dump. The SYS1.LOGREC entries and SVC dump should be examined to determine the primary error that occurred.

You might find the items listed in "Diagnostics" on page 993 useful in resolving the problem. In addition, see the description of reason code X'00D10252' for details of the information recorded in the variable recording area (VRA) of the system diagnostic work area (SDWA).

If you cannot resolve the problem, contact Support.

00D10252

This reason code is used to define the format of the information recorded in the variable recording area (VRA) of the system diagnostic work area (SDWA).

System action

An execution unit writes a record to SYS1.LOGREC and requests an SVC dump.

System programmer response

Obtain the SYS1.LOGREC and SVC dump.

You might find the items listed in “Diagnostics” on page 993 useful in resolving the problem. In addition, the following information is contained in the VRA of the SDWA:

- Reason code X'00D10252' stored with VRA key 6.
- The log buffer writer recovery tracking area is stored with VRA key 10.

00D10253

An application program check occurred in an MVCP instruction that attempted to move a parameter list or other data from the caller's address space to the queue manager address space.

System action

An execution unit writes a record to SYS1.LOGREC and requests an SVC dump.

System programmer response

Obtain the SYS1.LOGREC and SVC dump. You might find the items listed in “Diagnostics” on page 993 useful in resolving the problem.

Examine the area from which data was to be moved. It might be in the wrong key, or the address might be the cause of the problem. The incorrect instruction has a DA opcode and indicates the registers showing address and length to be moved.

00D10254

An application program check occurred in an MVCS instruction that attempted to move data from the queue manager address space to the caller's address space.

System action

An execution unit writes a record to SYS1.LOGREC and requests an SVC dump.

System programmer response

Obtain the SYS1.LOGREC and SVC dump. You might find the items listed in “Diagnostics” on page 993 useful in resolving the problem.

Examine the area to which data was to be moved. It might be in the wrong key, or the address might be the cause of the problem. The incorrect instruction has a DB opcode and indicates the registers showing address and length to be moved.

00D10257

The log RBA has reached or exceeded the value FFF800000000 (if 6-byte log RBAs are in use) or FFFFFFC000000000 (if 8-byte log RBAs are in use). The queue manager is terminated because the log RBA range has reached a CRITICAL level where the available range is too small for the queue manager to continue.

System action

The queue manager terminates with reason code 00D10257.

System programmer response

You need to reset the logs before the queue manager can be restarted. If you do not perform this action, the queue manager will abend once again after the next log data set switch.

For information on how to reset the logs using the CSQUTIL utility program, see RESETPAGE.

If your queue manager is using 6-byte log RBAs, consider converting the queue manager to use 8-byte log RBAs. See Planning to increase the maximum addressable log range for further information.

00D10261

While scanning the records and record segments in a log control interval (CI), it was discovered that the forward record chain was broken. This condition is the result of an incorrect record length in the log record header of some record in the log CI.

System action

This reason code can be issued by an active queue manager as the log buffers are scanned before they are written to the active log, or by the IBM MQ log services GET processor as a CI is retrieved from a user-specified active or archive log data set.

If the reason code is issued by an active queue manager, a diagnostic record is written to SYS1.LOGREC, and an SVC dump is requested.

- If the error was detected by CSQJOFF1, the archiving of the active log data set is terminated and the faulty active log data set is marked 'stopped'
- If the error was detected by CSQJR005, message CSQJ012E is issued and the calling agent is terminated
- If the error was detected by CSQJW009, message CSQJ012E is issued and the queue manager is terminated
- If the error was detected by CSQJW107, the queue manager is terminated

If this reason code is issued as the result of IBM MQ log services GET processing, no error is issued and no information is written to the SYS1.LOGREC data set.

System programmer response

You might find the items listed in "Diagnostics" on page 993 useful in resolving the problem. If you are unable to solve the problem, contact IBM Support.

00D10262

While scanning a log control interval (CI), the offset to the last record or record segment in the CI was found to be incorrect.

System action

This reason code can be issued by an active queue manager as the log buffers are scanned before they are written to the active log, or by the IBM MQ log services GET processor as a CI is retrieved from a user-specified active or archive log data set.

If the reason code is issued by an active queue manager, a diagnostic record is written to SYS1.LOGREC, and an SVC dump is requested.

- If the error was detected by CSQJOFF1, the archiving of the active log data set is terminated and the faulty active log data set is marked 'stopped'
- If the error was detected by CSQJR005, message CSQJ012E is issued and the calling agent is terminated
- If the error was detected by CSQJW009, message CSQJ012E is issued and the queue manager is terminated
- If the error was detected by CSQJW107, the queue manager is terminated

If this reason code is issued as the result of IBM MQ log services GET processing, no error is issued, and no information is written to the SYS1.LOGREC data set.

System programmer response

You might find the items listed in "Diagnostics" on page 993 useful in resolving the problem. If you are unable to solve the problem, contact IBM Support.

00D10263

While scanning a log control interval (CI), the VSAM RDF/CIDF control information was found to be incorrect.

System action

This reason code can be issued by an active queue manager as the log buffers are scanned before they are written to the active log, or by the IBM MQ log services GET processor as a CI is retrieved from a user-specified active or archive log data set.

If the reason code is issued by an active queue manager, a diagnostic record is written to SYS1.LOGREC, and an SVC dump is requested.

- If the error was detected by CSQJOFF1, the archiving of the active log data set is terminated and the faulty active log data set is marked 'stopped'
- If the error was detected by CSQJR005, message CSQJ012E is issued and the calling agent is terminated
- If the error was detected by CSQJW009, message CSQJ012E is issued and the queue manager is terminated
- If the error was detected by CSQJW107, the queue manager is terminated

If this reason code is issued as the result of IBM MQ log services GET processing, no error is issued, and no information is written to the SYS1.LOGREC data set.

System programmer response

You might find the items listed in "Diagnostics" on page 993 useful in resolving the problem. If you are unable to solve the problem, contact IBM Support.

00D10264

While scanning a log control interval (CI), the beginning log RBA of the CI was not the expected RBA.

System action

This reason code can be issued by an active queue manager as the log buffers are scanned before they are written to the active log, or by the IBM MQ log services GET processor as a CI is retrieved from a user-specified active or archive log data set.

If the reason code is issued by an active queue manager, a diagnostic record is written to SYS1.LOGREC, and an SVC dump is requested.

- If the error was detected by CSQJOFF1, the archiving of the active log data set is terminated and the faulty active log data set is marked 'stopped'
- If the error was detected by CSQJR005, message CSQJ012E is issued and the calling agent is terminated
- If the error was detected by CSQJW009, message CSQJ012E is issued and the queue manager is terminated
- If the error was detected by CSQJW107, the queue manager is terminated

If this reason code is issued as the result of IBM MQ log services GET processing, no error is issued, and no information is written to the SYS1.LOGREC data set.

System programmer response

You might find the items listed in "Diagnostics" on page 993 useful in resolving the problem. If you are unable to solve the problem, contact IBM Support.

00D10265

While scanning the records and record segments in a log control interval (CI), it was discovered that the backward record chain was broken. This condition is the result of an incorrect record length in the log record header of some record in the log CI.

System action

This reason code can be issued by an active queue manager as the log buffers are scanned before they are written to the active log, or by the IBM MQ log services GET processor as a CI is retrieved from a user-specified active or archive log data set.

If the reason code is issued by an active queue manager, a diagnostic record is written to SYS1.LOGREC, and an SVC dump is requested.

- If the error was detected by CSQJOFF1, the archiving of the active log data set is terminated
- If the error was detected by CSQJR005, message CSQJ012E is issued and the calling agent is terminated
- If the error was detected by CSQJW009, message CSQJ012E is issued and the queue manager is terminated
- If the error was detected by CSQJW107, the queue manager is terminated

If this reason code is issued as the result of IBM MQ log services GET processing, no error is issued, and no information is written to SYS1.LOGREC data set.

System programmer response

You might find the items listed in "Diagnostics" on page 993 useful in resolving the problem. If you are unable to solve the problem, contact IBM Support.

00D10266

While scanning a log control interval (CI), a unit of recovery ID or LINK RBA in some record was found to be inconsistent with the beginning log RBA of the CI.

System action

This reason code can be issued by an active queue manager as the log buffers are scanned before they are written to the active log, or by the IBM MQ log services GET processor as a CI is retrieved from a user-specified active or archive log data set.

If the reason code is issued by an active queue manager, a diagnostic record is written to SYS1.LOGREC, and an SVC dump is requested.

- If the error was detected by CSQJOFF1, the archiving of the active log data set is terminated and the faulty active log data set is marked 'stopped'
- If the error was detected by CSQJR005, message CSQJ012E is issued and the calling agent is terminated
- If the error was detected by CSQJW009, message CSQJ012E is issued and the queue manager is terminated
- If the error was detected by CSQJW107, the queue manager is terminated

If this reason code is issued as the result of IBM MQ log services GET processing, no error is issued, and no information is written to SYS1.LOGREC data set.

System programmer response

You might find the items listed in "Diagnostics" on page 993 useful in resolving the problem. If you are unable to solve the problem, contact IBM Support.

00D10267

While scanning a log control interval (CI), a middle or last spanned record segment was not the first segment contained in the log CI.

System action

This reason code can be issued by an active queue manager because the log buffers are scanned before they are written to the active log, or by the IBM MQ log services GET processor because a CI is retrieved from a user-specified active or archive log data set.

If the reason code is issued by an active queue manager, a diagnostic record is written to SYS1.LOGREC, and an SVC dump is requested.

- If the error was detected by CSQJOFF1, the archiving of the active log data set is terminated and the faulty active log data set is marked 'stopped'

- If the error was detected by CSQJR005, message CSQJ012E is issued and the calling agent is terminated
- If the error was detected by CSQJW009, message CSQJ012E is issued and the queue manager is terminated
- If the error was detected by CSQJW107, the queue manager is terminated

If this reason code is issued as the result of IBM MQ log services GET processing, no error is issued, and no information is written to the SYS1.LOGREC data set.

System programmer response

You might find the items listed in "Diagnostics" on page 993 useful in resolving the problem. If you are unable to solve the problem, contact IBM Support.

00D10268

While scanning a log control interval (CI), a first or middle spanned record segment was not the last segment contained in the log CI.

System action

This reason code can be issued by an active queue manager as the log buffers are scanned before they are written to the active log, or by the IBM MQ log services GET processor as a CI is retrieved from a user-specified active or archive log data set.

If the reason code is issued by an active queue manager, then a diagnostic record is written to SYS1.LOGREC, and an SVC dump is requested.

- If the error was detected by CSQJOFF1, the archiving of the active log data set is terminated and the faulty active log data set is marked 'stopped'
- If the error was detected by CSQJR005, message CSQJ012E is issued and the calling agent is terminated
- If the error was detected by CSQJW009, message CSQJ012E is issued and the queue manager is terminated
- If the error was detected by CSQJW107, the queue manager is terminated

If this reason code is issued as the result of IBM MQ log services GET processing, no error is issued, and no information is written to the SYS1.LOGREC data set.

System programmer response

You might find the items listed in "Diagnostics" on page 993 useful in resolving the problem. If you are unable to solve the problem, contact IBM Support.

00D10269

An unrecoverable error was found in one of the buffers, while moving the current log buffer to the static write buffer in preparation for the physical write to the active log.

System action

An execution unit writes a record to SYS1.LOGREC and requests an SVC dump. The queue manager then terminates.

System programmer response

You might find the items listed in "Diagnostics" on page 993 useful in resolving the problem. If you are unable to solve the problem, contact IBM Support.

00D10270

A LOG WRITE request completed unsuccessfully because the length of the log record header was not as expected. This is an internal error.

System action

An execution unit writes a record to SYS1.LOGREC and requests an SVC dump. The execution unit then terminates abnormally.

System programmer response

You might find the items listed in "Diagnostics" on page 993 useful in resolving the problem.

Examine the SYS1.LOGREC, console log, and SVC dump for information about prior errors during LOG WRITE processing.

If you are unable to solve the problem, contact IBM Support.

00D10327

A LOG READ completed unsuccessfully because of an invalid log LOGRBA. A log read, MODE(DIRECT) with a requested RBA does not match the start of a log record.

System action

An SVC dump is requested and the execution unit ends abnormally. If the log read error occurs during queue manager startup then the queue manager ends abnormally.

System programmer response

Log read with MODE(DIRECT) is most commonly used in the queue manager for verifying that the start RBA of a unit of work can be found on the log, before a sequential (maybe backward) read of the log data to recover locks on an in-doubt unit of work, or to back out a unit of work. It indicates that the queue manager is being started with incomplete log data available.

If you suspect an error in IBM MQ, collect the following data and contact IBM Support:

- The BSDS
- All active and archive logs
- The SVC dump created by this error

00D1032A

An unsuccessful completion of a LOG READ has occurred. BSDS does not map the specified RBA into a log data set. Either the BSDS is in error, or the log data set has been deleted.

System action

An execution unit writes a record to SYS1.LOGREC and requests an SVC dump. The execution unit then terminates abnormally.

System programmer response

You might find the items listed in "Diagnostics" on page 993 useful in resolving the problem.

00D1032B

Completion of a LOG READ was unsuccessful, because an error occurred while attempting to allocate a log data set.

System action

An execution unit writes a record to SYS1.LOGREC and requests an SVC dump. The execution unit then terminates abnormally.

System programmer response

You might find the items listed in "Diagnostics" on page 993 useful in resolving the problem.

Examine LOGREC and SVC dump information. Also, examine any prior messages with a CSQJ prefix from recovery log manager allocation processing.

00D1032C

A LOG READ completed unsuccessfully, because an error occurred while opening or closing a log data set.

System action

An execution unit writes a record to SYS1.LOGREC and requests an SVC dump. The execution unit then terminates abnormally.

System programmer response

You might find the items listed in "Diagnostics" on page 993 useful in resolving the problem.

Examine LOGREC and SVC dump information. Also, examine prior messages from recovery log manager open/close processing. These messages have a prefix of CSQJ.

00D1032E

A LOG READ completed unsuccessfully due to an internal error.

System action

An execution unit writes a record to SYS1.LOGREC and requests an SVC dump. The execution unit then terminates abnormally.

System programmer response

You might find the items listed in "Diagnostics" on page 993 useful in resolving the problem. Examine the SYS1.LOGREC and SVC dump information.

00D10340

An unsuccessful completion of a LOG READ has occurred. This reflects an internal recovery log manager (RLM) logic error.

System action

An execution unit writes a record to SYS1.LOGREC and requests an SVC dump. The execution unit then terminates abnormally.

System programmer response

You might find the items listed in "Diagnostics" on page 993 useful in resolving the problem.

Examine the SYS1.LOGREC, console log and SVC dump for information about prior errors during LOG READ processing.

If you cannot solve the problem, contact IBM Support.

00D10341

A LOG READ completed unsuccessfully because an error was detected during a Forward READ of the log record. This is an internal error.

System action

An execution unit writes a record to SYS1.LOGREC and requests an SVC dump. The execution unit then terminates abnormally.

System programmer response

You might find the items listed in "Diagnostics" on page 993 useful in resolving the problem.

Examine the SYS1.LOGREC, console log and SVC dump for information about prior errors during LOG READ processing.

If you cannot solve the problem, contact IBM Support.

00D10342

A LOG READ completed unsuccessfully because an error was detected during a backward READ of a log record. This is an internal error.

System action

An execution unit writes a record to SYS1.LOGREC and requests an SVC dump. The execution unit then terminates abnormally.

System programmer response

You might find the items listed in "Diagnostics" on page 993 useful in resolving the problem.

Examine the SYS1.LOGREC, console log and SVC dump for information about prior errors during LOG READ processing.

If you cannot solve the problem, contact IBM Support.

00D10343

A LOG READ completed unsuccessfully because an error was detected during a READ of a log record due to an invalid CI offset. This is an internal error.

System action

An execution unit writes a record to SYS1.LOGREC and requests an SVC dump. The execution unit then terminates abnormally.

System programmer response

You might find the items listed in "Diagnostics" on page 993 useful in resolving the problem.

Examine the SYS1.LOGREC, console log and SVC dump for information about prior errors during LOG READ processing.

If you cannot solve the problem, contact IBM Support.

00D10345

A LOG READ completed unsuccessfully because an error was received from a CATALOG LOCATE request for an archive log data set. The requested archive log data set might have been uncataloged or deleted.

System action

An execution unit writes a record to SYS1.LOGREC and requests an SVC dump. The execution unit then terminates abnormally.

System programmer response

You might find the items listed in "Diagnostics" on page 993 useful in resolving the problem. Examine the SYS1.LOGREC and SVC dump.

00D10348

The maximum retry count was exceeded while attempting to read a log RBA.

System action

An execution unit writes a record to SYS1.LOGREC and requests an SVC dump. The execution unit then terminates abnormally.

System programmer response

Check the console log for related errors. This problem might occur if the user has specified an archive or active log data set to the BSDS with an incorrect RBA range.

If you cannot solve the problem, contact IBM Support.

00D10354

A LOG READ request completed successfully but the length of the log record header was not as expected. This is an internal error.

System action

An execution unit writes a record to SYS1.LOGREC and requests an SVC dump. The execution unit then terminates abnormally.

System programmer response

You might find the items listed in "Diagnostics" on page 993 useful in resolving the problem.

Examine the SYS1.LOGREC, console log, and SVC dump for information about prior errors during LOG READ processing.

If you are unable to solve the problem, contact IBM Support.

00D10406

The bootstrap data set access service received a request with an invalid function code.

System action

An execution unit writes a record to SYS1.LOGREC and requests an SVC dump. The execution unit then terminates abnormally.

System programmer response

You might find the items listed in "Diagnostics" on page 993 useful in resolving the problem. If you cannot solve the problem, contact IBM Support.

00D10410

An unsuccessful completion of a READ BSDS RECORD has occurred. An error has been returned from VSAM.

System action

An execution unit writes a record to SYS1.LOGREC and requests an SVC dump. The execution unit then terminates abnormally.

System programmer response

Check the console log for return codes from VSAM.

If you are unable to resolve the problem, note these values, collect the items listed in "Diagnostics" on page 993, and contact IBM Support.

00D10411

An unsuccessful completion of a WRITE UPDATE BSDS RECORD has occurred. An error has been returned from VSAM.

System action

An execution unit writes a record to SYS1.LOGREC and requests an SVC dump. The execution unit then terminates abnormally.

System programmer response

Check the console log for return codes from VSAM.

If you are unable to resolve the problem, note these values, collect the items listed in "Diagnostics" on page 993, and contact IBM Support.

00D10412

An unsuccessful completion of a WRITE INSERT BSDS RECORD has occurred. An error has been returned from VSAM.

System action

An execution unit writes a record to SYS1.LOGREC and requests an SVC dump. The execution unit then terminates abnormally.

System programmer response

Check the console log for return codes from VSAM.

If you are unable to resolve the problem, note these values, collect the items listed in “Diagnostics” on page 993, and contact IBM Support.

00D10413

An unsuccessful completion of a DELETE BSDS RECORD has occurred. An error has been returned from VSAM.

System action

An execution unit writes a record to SYS1.LOGREC and requests an SVC dump. The execution unit then terminates abnormally.

System programmer response

Check the console log for return codes from VSAM.

If you are unable to resolve the problem, note these values, collect the items listed in “Diagnostics” on page 993, and contact IBM Support.

00D10419

An error was returned from the z/OS GETDSAB service.

System action

The current utility terminates abnormally.

System programmer response

Contact IBM Support.

00D1041A

An error was returned from the z/OS SWAREQ service.

System action

The current utility terminates abnormally.

System programmer response

Contact IBM Support.

00D1041B

The Db2 subsystem that a utility has connected to does not meet the minimum system requirements for this version of IBM MQ for z/OS.

System action

The current utility terminates abnormally.

System programmer response

Ensure that the Db2 data-sharing group name, and subsystem ID provided in the parameters to the utility are correct, and that the Db2 subsystem meets the system requirements for this version of IBM MQ for z/OS.

See IBM MQ prerequisites for a link to the IBM MQ for z/OS requirements web page.

00D10700

An error completion code was returned by SETLOCK OBTAIN.

System action

An execution unit writes a record to SYS1.LOGREC and requests an SVC dump. The execution unit then terminates abnormally.

System programmer response

You might find the items listed in “Diagnostics” on page 993 useful in resolving the problem. In the dump, register 0 contains the return code from SETLOCK OBTAIN.

00D10701

An error completion code was returned by SETLOCK RELEASE.

System action

An execution unit writes a record to SYS1.LOGREC and requests an SVC dump. The execution unit then terminates abnormally.

System programmer response

You might find the items listed in “Diagnostics” on page 993 useful in resolving the problem. In the dump, register 0 contains the return code from SETLOCK RELEASE.

00D10901

During recovery of a peer queue manager, the SCA member record of the peer queue manager cannot be found.

System action

This typically results in recovery of the peer queue manager being unable to continue.

System programmer response

Manually restarting the peer queue manager, which then attempts to resolve the problem.

00D10905

During recovery of a peer queue manager, a discrepancy has been detected between the peer queue manager's BSDS system timestamps. This can only occur if the peer queue manager was using dual BSDS.

System action

This typically results in recovery of the peer queue manager being unable to continue.

System programmer response

The print log map utility (CSQJU004) can be used to confirm the system timestamp discrepancy of the peer queue manager.

You can resolve the discrepancy by manually restarting the peer queue manager, which then attempts to resolve the problem.

Related tasks

Collecting troubleshooting information

z/OS *Lock manager codes (X'D3')*

If a lock manager reason code occurs that is not listed here, an internal error has occurred. Collect the following diagnostic items and contact IBM Support.

Diagnostics

- A description of the actions that led to the error or, if applicable, either a listing of the application program or the input string to a utility program that was being run at the time of the error.
- The console output for the period leading up to the error.
- The queue manager job log.
- The system dump resulting from the error.
- If you are using CICS, the CICS transaction dump output.

- Appropriate IBM MQ, z/OS, Db2, CICS, and IMS service levels.

00D301F1

An attempt to obtain storage was unsuccessful. This is probably because there is insufficient storage in your region.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Check that you are running in a region that is large enough. If not, reset your system and restart the queue manager. If this is not the cause of the problem, collect the items listed in “Diagnostics” on page 1013 and contact IBM Support.

00D301F2

An attempt to obtain storage was unsuccessful. This is probably because there is insufficient storage in your region.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Check that you are running in a region that is large enough. If not, reset your system and restart the queue manager. If this is not the cause of the problem, collect the items listed in “Diagnostics” on page 1013 and contact IBM Support.

00D301F3

An attempt to obtain storage was unsuccessful. This is probably because there is insufficient storage in your region.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Check that you are running in a region that is large enough. If not, reset your system and restart the queue manager. If this is not the cause of the problem, collect the items listed in “Diagnostics” on page 1013 and contact IBM Support.

00D301F4

An attempt to obtain storage was unsuccessful. This is probably because there is insufficient storage in your region.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Check that you are running in a region that is large enough. If not, reset your system and restart the queue manager. If this is not the cause of the problem, collect the items listed in “Diagnostics” on page 1013 and contact IBM Support.

00D301F5

An attempt to obtain storage was unsuccessful. This is probably because there is insufficient storage in your region.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Check that you are running in a region that is large enough. If not, reset your system and restart the queue manager. If this is not the cause of the problem, collect the items listed in “Diagnostics” on page 1013 and contact IBM Support.

00D302F1, 00D302F2, 00D302F3, 00D302F4, 00D302F5, 00D303F1, 00D303F2, 00D303F3, 00D304F1, 00D305F1, 00D306F1

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1013 and contact IBM Support.

00D31094, 00D31095, 00D31096, 00D31097

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'. In some cases, the queue manager might terminate with completion code X'6C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1013 and contact IBM Support.

Related tasks

Collecting troubleshooting information

z/OS *Message manager codes (X'D4')*

If a message manager reason code occurs that is not listed here, an internal error has occurred. Collect the following diagnostic items and contact IBM Support.

Diagnostics

- A description of the actions that led to the error or, if applicable, either a listing of the application program or the input string to a utility program that was being run at the time of the error.
- The console output for the period leading up to the error.
- The queue manager job log.
- The system dump resulting from the error.
- If you are using CICS , the CICS transaction dump output.
- Appropriate IBM MQ, z/OS, Db2, CICS, and IMS service levels.
- If you are using the IBM MQ Operations and Control panels, the ISPF panel name.

00D40001, 00D40002

An internal error has occurred while processing a command.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015 and contact IBM Support.

00D40003, 00D40004, 00D40007

An internal error has occurred while processing a DEFINE or ALTER command for a queue.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015 and contact IBM Support.

00D40008

An internal error has occurred while processing a DEFINE or ALTER command for a process.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015 and contact IBM Support.

00D40009

An internal error has occurred while processing a DEFINE or ALTER command for a queue.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015 and contact IBM Support.

00D4000A, 00D4000B, 00D4000C

An internal error has occurred while processing a command.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015 and contact IBM Support.

00D4000D

An internal error has occurred while attempting to establish a processing environment for the command processors.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015 and contact IBM Support.

00D4000E, 00D4000F

An internal error has occurred while attempting to establish a processing environment.

System action

The current execution unit terminates with completion code X'5C6'

System programmer response

Collect the items listed in “Diagnostics” on page 1015 and contact IBM Support.

00D40010

An internal error has occurred while processing a command.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015 and contact IBM Support.

00D40011, 00D40012, 00D40013, 00D40014

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015 and contact IBM Support.

00D40015

An attempt to write a trigger message to the initiation queue or the dead-letter queue was unsuccessful because of an internal error (for example, a storage overwrite).

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015 and contact IBM Support.

00D40016, 00D40017, 00D40018, 00D4001A, 00D4001B, 00D4001C, 00D4001D, 00D4001E, 00D4001F

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015 and contact IBM Support.

00D40020, 00D40021, 00D40022, 00D40023, 00D40024, 00D40025

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015 and contact IBM Support.

00D40026

An internal error has occurred while processing a DEFINE CHANNEL or ALTER command for a channel.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015 and contact IBM Support.

00D40027, 00D40028, 00D40029, 00D4002A, 00D4002B, 00D4002C

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015 and contact IBM Support.

00D4002D

An attempt to write a message to a queue was unsuccessful because of an internal error (for example, a storage overwrite).

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015 and contact IBM Support.

00D4002E

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'

System programmer response

Collect the items listed in “Diagnostics” on page 1015 and contact IBM Support.

00D4002F

An internal error has occurred while processing a channel command.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015 and contact IBM Support.

00D40030

The report option requested in a message was not recognized.

System action

The current execution unit terminates with completion code X'5C6'. A dump is produced.

System programmer response

Correct the value of the report option field (the value specified is given in register 2).

00D40031, 00D40032

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015 and contact IBM Support.

00D40033

An internal error has occurred while processing a STGCLASS command.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015 and contact IBM Support.

00D40034, 00D40035, 00D40036, 00D40037, 00D40038, 00D40039

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015 and contact IBM Support.

00D4003B

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015. Also collect details of the queue sharing group (QSG) and of the queue managers active, as well as the queue managers defined to the queue sharing group at the time of the error. This information can be obtained by entering the following z/OS commands:

```
D XCF,GRP
```

to display a list of all QSGs in the coupling facility.

```
D XCF,GRP,qsg-name,ALL
```

to display status about the queue managers defined to queue sharing group qsg-name. contact IBM Support.

00D4003C, 00D4003D

An internal error has occurred while processing a DEFINE CFSTRUCT or ALTER CFSTRUCT or DELETE CFSTRUCT command.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015 and contact IBM Support.

00D4003E

An internal error has occurred while processing an AUTHINFO command.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015 and contact IBM Support.

00D4003F

An internal error has occurred while processing a DEFINE MAXSMSGS or ALTER QMGR command.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015 and contact IBM Support.

00D40040

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015 and contact IBM Support.

00D40042

An internal processing error has occurred. The repository cannot locate an object that it has been asked to release.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015 and contact IBM Support.

00D40043, 00D40044, 00D40045, 00D40046, 00D40047, 00D40048

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015 and contact IBM Support.

00D40049

An internal processing error has occurred while attempting to create the queue manager object during end restart processing.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015 and contact IBM Support.

00D40050

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'. The IGQ agent then attempts to recover.

System programmer response

If the IGQ agent fails to recover properly, an attempt could be made to disable the SYSTEM.QSG.TRANSMIT.QUEUE to force the IGQ agent to enter retry, or if this fails, the IGQ agent task can be restarted by issuing an ALTER QMGR IGQ(ENABLED) command or by restarting the queue manager.

Collect the items listed in “Diagnostics” on page 1015 and contact IBM Support.

00D40051, 00D40052

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015 and contact IBM Support.

00D40053

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015 together with a dump of the coupling facility list structure that the shared queue is defined to use, and contact IBM Support.

00D40054

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015. Also collect details of the queue sharing group (QSG) and of the queue managers active, as well as the queue managers defined to the queue sharing group at the time of the error. This information can be obtained by entering the following z/OS commands:

```
D XCF,GRP
```

to display a list of all QSGs in the coupling facility.

```
D XCF,GRP,qsg-name,ALL
```

to display status about the queue managers defined to queue sharing group qsg-name. contact IBM Support.

00D40055, 00D40056

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015 and contact IBM Support.

00D40060

While performing Shared Channel Recovery Processing, Db2 was found to be inactive.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Check why Db2 related tasks are unavailable.

The recovery process is terminated; some channels might have been recovered, while others have not. Any channels that were not recovered will be recovered when the recovery process next runs; alternatively, they can be restarted manually. For more information about recovery and restart mechanisms used by IBM MQ, see Recovery and restart.

00D40062, 00D40064, 00D40065, 00D40066

An internal error has occurred during shared channel recovery.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015 and contact IBM Support.

The recovery process is terminated; some channels may have been recovered, while others have not. Any channels that were not recovered will be recovered when the recovery process next runs; alternatively, they can be restarted manually. For more information about recovery and restart mechanisms used by IBM MQ, see Recovery and restart.

00D40067

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015 and contact IBM Support.

00D40068

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'. In some cases, the queue manager might terminate with completion code X'6C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015 and contact IBM Support.

Restart the queue manager if necessary.

00D40069

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015. Also collect details of the queue sharing group (QSG) and of the queue managers active, as well as the queue managers defined to the queue sharing group at the time of the error. This information can be obtained by entering the following z/OS commands:

```
D XCF,GRP
```

to display a list of all QSGs in the coupling facility.

```
D XCF,GRP,qsg-name,ALL
```

to display status about the queue managers defined to queue sharing group qsg-name. contact IBM Support.

00D40070

An internal error has occurred involving the cluster cache.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015 and the channel initiator job log, and contact IBM Support.

00D40071, 00D40072, 00D40073, 00D40074, 00D40075, 00D40076, 00D40077, 00D40078, 00D40079, 00D4007A, 00D4007B, 00D4007C, 00D4007D, 00D4007E, 00D4007F

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015 and contact IBM Support.

Restart the queue manager if necessary.

00D40080

An internal error has occurred invloving the cluster cache.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015 and the channel initiator job log, and contact IBM Support.

00D40081

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015 and contact IBM Support.

Restart the queue manager if necessary.

00D40082

An internal error has occurred invloving the cluster cache.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015 and the channel initiator job log, and contact IBM Support.

00D40083

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015 and contact IBM Support.

Restart the queue manager if necessary.

00D40084

An internal error has occurred when opening a managed destination queue.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015 and contact IBM Support.

Restart the queue manager if necessary.

00D40085

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015 and contact IBM Support.

Restart the queue manager if necessary.

00D40086, 00D40087

An internal error has occurred while processing a DEFINE or ALTER command for a subscription.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015 and contact IBM Support.

00D40091

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015 and contact IBM Support.

Restart the queue manager if necessary.

00D4009C

An internal error has occurred while processing an **ALTER SMDS** or **RESET SMDS** command.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015 and contact IBM Support.

00D4009D

An internal error has occurred while processing a **START SMDSCONN** or **STOP SMDSCONN** command.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015 and contact IBM Support.

00D400B1

While putting a message, an error was detected in the chaining of message headers.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Check the chaining fields (**CodedCharSetId**, **Encoding**, and **Format**) in the MQMD and headers for the problem message to determine which values are invalid or inconsistent.

At each point in the header chain, the field values must correctly describe the data in the next header:

- The **Format** field identifies the correct format of the next header
- The **CodedCharSetId** field identifies the character set of text fields in the next header
- The **Encoding** field identifies the numeric encoding of numeric fields in the next header

00D400B9

A cluster cache allocation request exceeded the maximum allowed size.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015 and contact IBM Support.

00D400D0

Unable to get below the line storage for data control blocks when attempting to open the QM INI (CSQMQMIN) data set.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015 and contact IBM Support.

V 10.0.0 **00D400D7**

An internal error occurred when attempting to read cached indoubt connection list entry mappings for an inactive peer queue manager.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015 and contact IBM Support.

00D401F1

Whilst processing a get message request, the specified search type (message identifier or correllation identifier) was found to be in error. This indicates a data corruption error.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1015 and contact IBM Support.

00D44001

This reason code is issued in message CSQM090E when a command has failed. This code indicates that an object of the specified name exists, but is of a different subtype; it might not necessarily have the same disposition in the queue sharing group. This can only occur with subtypes of queues or channels. Message CSQM099I is also issued, indicating the object in error.

Severity

8

System action

The command is ignored.

System programmer response

Reissue the command, ensuring that all object subtypes are correct.

00D44002

This reason code is issued in message CSQM090E when a command has failed. This code indicates that the object specified on the request could not be located. Message CSQM094I or message CSQM125I is also issued, indicating the object in error.

It is also issued in message CSQM086E, indicating that the queue manager object could not be located.

Severity

8

System action

For CSQM090E, the command is ignored. For CSQM086E, the queue manager fails to restart.

System programmer response

If you are using a queue sharing group, check that Db2 is available and not suspended. Define the object in question. For the queue manager, reissue the START QMGR command to restart the queue manager.

Note: An object of the same name and type, but of a different disposition, might already exist. If you are dealing with a queue or channel object, an object of the same name, but of a different subtype, might already exist.

00D44003

This reason code is issued in message CSQM090E when a command has failed. This code indicates that the object specified on the request already exists. This will only arise when trying to define a new object. Message CSQM095I is also issued.

Severity

8

System action

The command is ignored.

System programmer response

Use the object in question.

00D44004

This reason code is issued in message CSQM090E when a command has failed. This code indicates that one or more of the keywords on the command failed the parameter validation rules that apply to them. One or more other more specific messages are also issued, indicating the reason for the validation failure.

Severity

8

System action

The command is ignored.

System programmer response

Refer to the more specific associated message to determine what the error is.

00D44005

This reason code is issued in message CSQM090E when a command has failed. This code indicates that one of the following situations has occurred:

- The object specified on the request is currently open. This typically happens when an object is in use through the API or a trigger message is being written to it, but it could also arise because the object specified is in the process of being deleted. For a local queue, it can occur because there are messages currently on the queue. Message CSQM101I or CSQM115I is also issued.
- A request has been issued for a local queue, but this queue has incomplete units of recovery outstanding for it. Message CSQM110I is also issued.
- An alter, delete, or define request was made against a storage class that is in use (that is, there is a queue defined as using the storage class, and there are messages currently on the queue. Message CSQM101I is also issued.
- An ALTER CFSTRUCT command was issued and an associated shared queue has messages or uncommitted message activity.

Severity

8

System action

The command is ignored.

System programmer response

Refer to the description of message CSQM101I, CSQM110I, or CSQM115I as appropriate.

00D44006

This reason code is issued in message CSQM090E when a command has failed. This code indicates that a request has been issued to delete a local queue. The PURGE option has not been specified, but there are messages on the queue. Message CSQM103I is also issued.

Severity

8

System action

The command is ignored.

System programmer response

If the local queue must be deleted, even though there are messages on it, reissue the command with the PURGE option.

00D44007

This reason code is issued in message CSQM090E when a command has failed. This code indicates that a request has been issued for a local queue that is dynamic, but this queue has been flagged for deletion. Message CSQM104I is also issued.

Severity

8

System action

The command is ignored.

System programmer response

None, the local queue will be deleted as soon as possible.

00D44008

This reason code is issued in message CSQM090E when a command has failed. This code indicates that the object specified on the request needs updating because the IBM MQ version has changed, but that this cannot be done because the object is currently open. Message CSQM101I is also issued.

Severity

8

System action

The command is ignored.

System programmer response

Wait until the object is closed and reissue the command.

00D44009

This reason code is issued in message CSQM090E when a command has failed, and is accompanied by message CSQM112E or message CSQM117E indicating the object in error. It is also issued in message CSQM086E during queue manager restart.

This code indicates that a request has been issued for an object, but the object information could not be accessed because of an error on page set zero.

Severity

8

System action

The command is ignored or the queue manager fails to restart.

System programmer response

Check for error messages on the console log that might relate to the problem. Verify that page set zero is set up correctly; refer to Managing page sets for information about this.

00D4400A

This reason code is issued in message CSQM090E when a command has failed, and is accompanied by message CSQM113E indicating the object in error. It is also issued in message CSQM086E during queue manager restart. This code indicates that a request has been issued for an object, but page set zero is full.

Severity

8

System action

The command is ignored or the queue manager fails to restart.

System programmer response

Increase the size of page set zero. Refer to Managing page sets for information about how to do this.

00D4400B

This reason code is issued in message CSQM090E when a command has failed, and is accompanied by message CSQM114E. This code indicates that a request has been issued for a local queue, but no more local queues could be defined. There is an implementation limit of 524 287 for the total number

of local queues that can exist. For shared queues, there is a limit of 512 queues in a single coupling facility structure, and 512 structures altogether.

For the indexes used by shared queues (OBJ_QUEUE_IX1 and OBJ_QUEUE_IX2), 48 KB of space allocation is sufficient for 512 queues.

Severity

4

System action

The command is ignored.

System programmer response

Delete any existing queues that are no longer required.

00D4400C

This reason code is issued in message CSQM090E when a command has failed. It indicates that the command is not allowed for a particular subtype of an object, as shown in the accompanying more specific message.

Severity

4

System action

The command is ignored.

System programmer response

Reissue the command with the object name specified correctly.

00D4400D

This reason code is issued in message CSQM090E when a command has failed, and is accompanied by message CSQM127I. This code indicates that a request was issued specifying a namelist as a list of cluster names, but there are no names in the namelist.

Severity

8

System action

The command is ignored.

System programmer response

Specify a namelist that is not empty.

00D4400E

This reason code is issued in message CSQM090E when a command has failed, and is accompanied by message CSQM112E or message CSQM117E indicating the object in error. It is also issued in message CSQM086E during queue manager restart. This code indicates that a request has been issued for an object, but that a page set that it requires is not defined.

Severity

8

System action

The command is ignored or the queue manaager fails to restart.

System programmer response

Ensure that the necessary page set is defined in the initialization input data set CSQINP1, and has a DD statement in the queue manager started task JCL procedure. Restart the queue manager.

00D4400F

This reason code is issued in message CSQM090E when a command has failed, and is accompanied by message CSQM112E or message CSQM117E indicating the object in error. It is also issued in message CSQM086E during queue manager restart. This code indicates that a request has been issued for an object, but that a page set that it requires is not open.

Severity

8

System action

The command is ignored or the queue manager fails to restart.

System programmer response

Ensure that the necessary page set is defined in the initialization input data set CSQINP1, and has a DD statement in the queue manager started task JCL procedure. Restart the queue manager.

00D44010

This reason code is issued in message CSQM090E when a command has failed. This code indicates that a request was issued to change the default transmission queue for the queue manager, but the queue is already in use.

Severity

8

System action

The command is ignored.

System programmer response

Wait until the queue is no longer in use, or choose another queue.

00D44011

This reason code is issued in message CSQM090E when a command has failed, and is accompanied by message CSQM128E. This code indicates that a request was issued that required a message to be sent to a command queue, but the message could not be put.

Severity

8

System action

The command is ignored.

System programmer response

Resolve the problem with the command queue.

00D44013

This reason code is issued in message CSQM090E when a command has failed, and is accompanied by message CSQM160I indicating the object in error.

Severity

8

System action

The command is ignored.

System programmer response

See message CSQM160I for more information.

00D44014

This reason code is issued in message CSQM090E when a command has failed, and is accompanied by message CSQM161I.

Severity

8

System action

The command is ignored.

System programmer response

See message CSQM161I for more information.

00D44015

This reason code is issued in message CSQM090E when a command has failed, and is accompanied by message CSQM164I indicating the object in error.

Severity

8

System action

The command is ignored.

System programmer response

See message CSQM164I for more information.

00D44016

This reason code is issued in message CSQM090E when a command has failed, and is accompanied by message CSQM163I indicating the object in error.

Severity

8

System action

The command stops processing.

System programmer response

See message CSQM163I for more information.

00D44017

This reason code is issued in message CSQM090E when a command has failed, and is accompanied by message CSQM112E or message CSQM117E indicating the object in error. It is also issued in message CSQM086E during queue manager restart.

This code indicates that a request has been issued for an object, but the object information could not be accessed because coupling facility structure has failed.

Severity

8

System action

The command is ignored or the queue manager fails to restart.

System programmer response

Check for error messages on the console log that might relate to the problem. Use the RECOVER CFSTRUCT command to recover the coupling facility structure.

00D44018

This reason code is issued in message CSQM090E when a command has failed, and is accompanied by message CSQM112E or message CSQM117E indicating the object in error. It is also issued in message CSQM086E during queue manager restart.

This code indicates that a request has been issued for an object, but the object information could not be accessed because there is an error or inconsistency in the coupling facility information.

This code might also occur as a result of the coupling facility structure for the queue being full.

Severity

8

System action

The command is ignored or the queue manager fails to restart.

System programmer response

Check for error messages on the console log that might relate to the problem.

Check that the coupling facility structure for the queue is not full.

Check that Db2 is available and not suspended. If the problem persists, it might be necessary to restart the queue manager.

00D44019

This reason code is issued in message CSQM090E when a command has failed, and is accompanied by message CSQM112E or message CSQM117E indicating the object in error. It is also issued in message CSQM086E during queue manager restart.

This code indicates that a request has been issued for an object, but the object information could not be accessed because Db2 is not available or is suspended.

Severity

8

System action

The command is ignored or the queue manager fails to restart.

System programmer response

Check for error messages on the console log that might relate to the problem. Check that Db2 is available and not suspended.

00D44023

This reason code is issued in message CSQM090E and is accompanied by message CSQM117E when a command cannot be executed because a CF structure is not available.

System action

The command is ignored.

System programmer response

See reason code “2346 (092A) (RC2346): MQRC_CF_STRUC_IN_USE” on page 1277 for more information.

00D4001B

This reason code is issued in message CSQM090E when a command has failed, and is accompanied by message CSQM182E.

Severity

8

System action

The command is ignored.

System programmer response

See message CSQM182E for more information.

00D4001C

This reason code is issued in message CSQM090E when a command has failed, and is accompanied by message CSQM183E.

Severity

8

System action

The command is ignored.

System programmer response

See message CSQM183E for more information.

00D4001D

This reason code is issued in message CSQM090E when a command has failed, and is accompanied by message CSQM185E.

Severity

8

System action

The command is ignored.

System programmer response

See message CSQM185E for more information.

00D4001E

This reason code is issued in message CSQM090E when a command has failed, and is accompanied by message CSQM186E.

Severity

8

System action

The command is ignored.

System programmer response

See message CSQM186E for more information.

00D4401F

This reason code is issued in message CSQM090E when a command has failed, and is accompanied by message CSQM190E.

Severity

8

System action

The command is ignored.

System programmer response

See message CSQM190E for more information.

00D44020

This reason code is issued in message CSQM090E when a PUBSUB command cannot be executed because PUBSUB is disabled.

System action

The command is ignored.

System programmer response

See message CSQM292I for more information.

00D4F001

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in "Diagnostics" on page 1015 and contact IBM Support.

Related tasks

Collecting troubleshooting information

z/OS *Command server codes (X'D5')*

If a command server reason code occurs that is not listed here, an internal error has occurred. Collect the following diagnostic items and contact IBM Support.

Diagnostics

- A description of the actions that led to the error or, if applicable, either a listing of the application program or the input string to a utility program that was being run at the time of the error.
- The console output for the period leading up to the error.
- The queue manager job log.
- The system dump resulting from the error.
- Appropriate IBM MQ, z/OS, Db2, CICS, and IMS service levels.
- Any trace information collected.
- If message CSQN104I or CSQN202I was issued was issued, return and reason codes from the message.

00D50101

During initialization, the command server was unable to obtain storage. This is probably because there is insufficient storage in your region.

System action

Message CSQN104I is sent to the console containing this reason code and the return code from the internal storage macro. None of the commands in the initialization data set currently being processed are performed. Queue manager startup continues.

Note: If there is a storage problem, startup might not be successful.

System programmer response

Check that you are running in a region that is large enough, and if not, reset your system and restart the queue manager. If this is not the cause of the problem, collect the following items and contact IBM Support:

- Return and reason codes from CSQN104I message
- Trace of startup (if available)

00D50102

The command preprocessor ended abnormally while processing a command in the initialization input data set.

System action

Message CSQ9029E is produced, followed by message CSQN103I with this code as the return code, and a reason code of -1 indicating that the command was not processed, and a dump is produced. The next command is processed.

System programmer response

Look in the output data set to determine the command in error. Check that the command is correctly formed, that it applies to a valid object.

If the command is correct, collect the items listed in "Diagnostics" on page 1034 and contact IBM Support.

00D50103

During initialization, an internal error occurred.

System action

Message CSQN104I is sent to the z/OS console, indicating the return and reason codes from the internal macro. The command server stops, without processing any commands.

System programmer response

Review the job log for messages about other errors that might be related. If you are unable to solve the problem, collect the items listed in "Diagnostics" on page 1034, and contact IBM Support.

00D50104

An internal error occurred during initialization.

System action

Message CSQN104I is sent to the z/OS console, indicating the return and reason codes from the internal macro. The command server stops, without processing any commands.

System programmer response

Stop and restart the queue manager.

Collect the items listed in "Diagnostics" on page 1034 and contact IBM Support.

00D50105

An internal error has occurred.

System action

The command server terminates, and a dump is produced.

System programmer response

Stop and restart the queue manager.

Collect the items listed in "Diagnostics" on page 1034 and contact IBM Support.

00D50201

The command server was unable to obtain storage while starting. This return code typically occurs because there is insufficient storage in your region.

System action

Message CSQN202I is sent to the z/OS console, indicating the return code from the internal storage macro. The command server stops, without processing any commands.

System programmer response

Check that you are running in a region that is large enough, and if not, reset your system and restart the queue manager. If this is not the cause of the problem, collect the items listed in "Diagnostics" on page 1034 and contact IBM Support.

00D50202

An internal error has occurred.

System action

Message CSQN202I is sent to the z/OS console, indicating the return code from the internal macro. The command server stops, without processing any commands.

System programmer response

Review the job log for messages about other errors that might be related. If you are unable to solve the problem, collect the items listed in "Diagnostics" on page 1034 and contact IBM Support.

00D50203

An internal error has occurred.

System action

Message CSQN202I is sent to the z/OS console, indicating the return code from the internal macro. The command server stops, without processing any commands.

System programmer response

Issue the START CMDSERV command to restart the command server.

Collect the items listed in "Diagnostics" on page 1034 and contact IBM Support.

00D50208

The command server was unable to obtain storage during startup.

System action

Message CSQN202I is sent to the z/OS console, indicating the return code from the internal macro. The command server stops, without processing any commands.

System programmer response

Check that you are running in a region that is large enough, and if not, reset your system and restart the queue manager. If this is not the cause of the problem, collect the items listed in "Diagnostics" on page 1034 and contact IBM Support.

00D50209

The command preprocessor ended abnormally while processing a command from the command server.

System action

Message CSQN205I is put onto the reply-to queue with COUNT=1, RETURN=00D50209, and REASON=-1 indicating that the command has not been processed. The command server processes the next command.

System programmer response

Check that the command is correctly formed, that it applies to a valid object.

If the command is correct, collect the items listed in "Diagnostics" on page 1034 and contact IBM Support.

00D5020C

While waiting for a command, the command server did not recognize the reason for the end of the wait. This is because it was not one of the following:

- The arrival of a message
- The **STOP CMDSERV** command

System action

Messages CSQN203I and CSQN206I are sent to the console, containing the return and reason codes from the request function, and the ECB list.

The command server is terminated and a dump is produced.

System programmer response

Issue the **START CMDSERV** command to restart the command server.

Collect the items listed in "Diagnostics" on page 1034 and contact IBM Support.

00D5020E

The command processor attempted to get a command from the system-command-input queue, but the attempt was unsuccessful because of an internal error.

System action

The command server continues processing. Message CSQN203I is written to the console containing the return and reason codes from the API call.

System programmer response

Collect the items listed in "Diagnostics" on page 1034 and contact IBM Support.

00D5020F

The command processor got a command from the system-command-input queue, but was unable to process it because the message was not of type MQMT_REQUEST.

System action

The command processor processes the next command message.

00D50210

The command processor got a command from the system-command-input queue, but was unable to process it because the command message was of length zero.

System action

The command processor processes the next command message.

00D50211

The command processor got a command from the system-command-input queue, but was unable to process it because the command message consisted of blank characters only.

System action

The command processor processes the next command message.

00D50212

The command processor got a command from the system-command-input queue, but was unable to process it because the command message was greater than 32 762 characters long.

System action

The command processor processes the next command message.

00D54000

An internal error has occurred.

System action

The command server is terminated and a dump is produced.

System programmer response

Issue the **START CMDSERV** command to restart the command server.

Collect the items listed in "Diagnostics" on page 1034 and contact IBM Support.

00D54nnn

The command processor got a command from the system-command-input queue, but was unable to process it because the command message indicated that data conversion was required and an error occurred during conversion. *nnn* is the reason code (in hexadecimal) returned by the MQGET call.

System action

The command processor processes the next command message.

System programmer response

Refer to "API completion and reason codes" on page 1151 for information about the reason code *nnn*.

Related tasks

Collecting troubleshooting information

z/OS *Buffer manager codes (X'D7')*

If a buffer manager reason code occurs that is not listed here, an internal error has occurred. Collect the following diagnostic items and contact IBM Support.

Diagnostics

- A description of the actions that led to the error or, if applicable, either a listing of the application program or the input string to a utility program that was being run at the time of the error.
- The console output for the period leading up to the error.
- The queue manager job log.
- The IBM MQ active log data set.
- The system dump resulting from the error.
- If you are using CICS , the CICS transaction dump output.
- Appropriate IBM MQ, z/OS, Db2, CICS, and IMS service levels.

00D70101

An attempt to obtain storage for a buffer manager control block (the PANC) was unsuccessful. This is probably because there is insufficient storage in your region.

System action

The queue manager is terminated, an entry is written to SYS1.LOGREC, and a dump is produced. Registers 2 and 0 contain the return and reason codes from the STORAGE or GETMAIN request.

System programmer response

Check that you are running in a region that is large enough, and if not, reset your system and restart the queue manager. If this does not resolve the problem, note the register values, and contact IBM Support.

00D70102

The name of the queue manager being restarted does not match the name recorded in a prior checkpoint log record.

System action

The queue manager is terminated, an entry is written to SYS1.LOGREC, and a dump is produced. This is preceded by message CSQP006I. Register 0 contains the name found in the log record. Register 2 contains the name of the queue manager being restarted.

System programmer response

Change the started task JCL procedure xxxxMSTR for the queue manager to name the appropriate bootstrap and log data sets.

The print log utility, CSQ1LOGP, can be used to view checkpoint records. You might also find the MQ active log data set useful for problem determination.

00D70103

An attempt to obtain storage for a buffer manager control block (a PSET) was unsuccessful.

System action

The queue manager is terminated, an entry is written to SYS1.LOGREC, and a dump is produced. Registers 2 and 0 contain the return and reason codes from the STORAGE or GETMAIN request.

System programmer response

Restart the queue manager.

Note the register values, and contact IBM Support.

00D70104

An attempt to obtain storage for a buffer manager control block (a BHDR) was unsuccessful.

System action

The queue manager is terminated, an entry is written to SYS1.LOGREC, and a dump is produced. Registers 2 and 0 contain the return and reason codes from the STORAGE or GETMAIN request.

System programmer response

Restart the queue manager.

Note the register values, and contact IBM Support.

00D70105

An internal error has occurred during dynamic page set expansion.

System action

The current page set extend task is terminated, an entry is written to SYS1.LOGREC, and a dump is produced. No further attempt will be made to expand the page set until the queue manager is restarted. Subsequent dynamic page set extend requests for other page sets are processed.

System programmer response

Collect the items listed in “Diagnostics” on page 1038 and contact IBM Support.

00D70106

An internal error has occurred.

System action

An entry is written to SYS1.LOGREC, and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 1038 and contact IBM Support.

00D70108

An attempt to obtain storage for the buffer pool was unsuccessful.

System action

The queue manager is terminated, an entry is written to SYS1.LOGREC, and a dump is produced. Register 2 contains the return code from the STORAGE, GETMAIN or IARV64 GETSTOR request. Register 3 contains the buffer pool number.

System programmer response

Provide sufficient storage for the number of buffers specified in the DEFINE BUFFPOOL command.

If the buffer pool is backed by page fixed storage, that is it has a PAGECLAS of FIXED4KB, check that there is enough real storage available on the system. For more information, see Address space storage.

If it is not possible to rectify the problem:

- Alter the definition of the buffer pool in the CSQINP1 data set, to include the REPLACE attribute, and specify a smaller number of buffers, or
- Change the PAGECLAS attribute to 4KB.

00D7010A

An internal storage error has occurred.

System action

The queue manager is terminated, an entry is written to SYS1.LOGREC, and a dump is produced. Registers 2 and 3 contain the return and reason codes from the IARV64 GETSTOR request. Register 4 contains the buffer pool number.

System programmer response

Increase the value of the MEMLIMIT parameter.

00D70112

A critical process could not be started during queue manager initialization. This could be because there is insufficient storage in your region.

System action

The queue manager is terminated, an entry is written to SYS1.LOGREC, and a dump is produced. Register 0 contains the reason code for the error.

System programmer response

Check that you are running in a region that is large enough. If not, reset your system and restart the queue manager. If this does not resolve the problem, note the completion code and the reason code and contact IBM Support.

00D70113

A critical process could not be started during queue manager initialization. This could be because there is insufficient storage in your region.

System action

The queue manager is terminated, an entry is written to SYS1.LOGREC, and a dump is produced. Register 0 contains the reason code for the error.

System programmer response

Check that you are running in a region that is large enough. If not, reset your system and restart the queue manager. If this does not resolve the problem, note the completion code and the reason code and contact IBM Support.

00D70114

An internal cross-component consistency check failed.

System action

The request is terminated, an entry is written to SYS1.LOGREC, and a dump is produced. Register 0 contains the value in error.

System programmer response

Note the completion code and the reason code, collect the MQ active log data set, and contact IBM Support.

00D70116

An I/O error has occurred.

System action

An entry is written to SYS1.LOGREC, and a dump is produced. Register 0 contains the Media Manager reason code from an MMCALL call. In some circumstances, the queue manager will terminate. (This depends on the nature of the error, and the page set on which the error occurred.)

System programmer response

Restart the queue manager if necessary.

See the *z/OS DFSMSdfp Diagnosis* manual for information about return codes from the Media Manager. If you do not have access to the required manual, contact IBM Support, quoting the Media Manager reason code.

You might also find the MQ active log data set useful for problem determination.

00D70118

A page was about to be written to a page set, but was found to have improper format. The executing thread is terminated. (If this is the deferred write processor, the queue manager is terminated)

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Restart the queue manager. If the problem persists collect the items listed in “Diagnostics” on page 1038 and contact IBM Support.

00D70120

No buffers are available to steal. An executing thread needed a buffer in a buffer pool to bring a page in from the page set. The buffer pool is over-committed, and despite attempts to make more buffers available, including writing pages to the page set, no buffers could be released.

System action

The current execution unit terminates with completion code X'5C6'. The API request is terminated with reason code MQRC_UNEXPECTED_ERROR, with the aim of reducing demand for the buffer pool.

System programmer response

Determine the problem buffer pool from preceding CSQP019I and CSQP020E messages. Review the size of the buffer pool with the DISPLAY USAGE command. Consider increasing the size of the buffer pool using the ALTER BUFFPOOL command.

00D70122

An unrecoverable error has occurred during check point.

System action

The queue manager is terminated, an entry is written to SYS1.LOGREC, and a dump is produced. Register 0 contains the reason code for the error.

System programmer response

Restart the queue manager.

Note the completion code and the reason code, collect the MQ active log data set, and contact IBM Support.

00D70133

An internal consistency check failed.

System action

The request is terminated, an entry is written to SYS1.LOGREC, and a dump is produced.

System programmer response

Note the completion code and the reason code, collect the MQ active log data set, and contact IBM Support.

00D70136

A critical process could not be started during queue manager initialization. This could be because there is insufficient storage in your region.

System action

The queue manager is terminated, an entry is written to SYS1.LOGREC, and a dump is produced. Register 0 contains the reason code for the error.

System programmer response

Check that you are running in a region that is large enough. If not, reset your system and restart the queue manager. If this does not resolve the problem, note the completion code and the reason code and contact IBM Support.

00D70137

A critical process could not be started during queue manager initialization. This could be because there is insufficient storage in your region.

System action

The queue manager is terminated, an entry is written to SYS1.LOGREC, and a dump is produced. Register 0 contains the reason code for the error.

System programmer response

Check that you are running in a region that is large enough. If not, reset your system and restart the queue manager. If this does not resolve the problem, note the completion code and the reason code and contact IBM Support.

00D70139

An attempt to allocate 64 bit storage for internal use failed.

System action

The queue manager is terminated.

System programmer response

Raise the value of the MEMLIMIT parameter. For more information, see Address space storage.

00D7013A

An attempt to allocate storage for internal use failed. Register 2 contains the return code from the STORAGE request.

System action

The queue manager is terminated.

System programmer response

Provide sufficient storage. For more information, see Address space storage.

00D7013B

An internal consistency check failed.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect buffer manager problem determination information, and contact IBM Support.

Related tasks

Collecting troubleshooting information

z/OS *Recovery manager codes (X'D9')*

If a recovery manager reason code occurs that is not listed here, an internal error has occurred. Collect the following diagnostic items and contact IBM Support.

Diagnostics

- A description of the actions that led to the error or, if applicable, either a listing of the application program or the input string to a utility program that was being run at the time of the error.
- The console output for the period leading up to the error.
- The queue manager job log.
- The system dump resulting from the error.
- A printout of SYS1.LOGREC.
- If you are using CICS , the CICS transaction dump output.
- Appropriate IBM MQ, z/OS, Db2, CICS, and IMS service levels.

00D90000

A recovery manager module received control from its FRR for retry and found an invalid retry point identifier. The name of the module in which the error occurred appears in the SYS1.LOGREC entry showing this reason code in register 15.

System action

Standard diagnostic information is provided. The error is recorded in SYS1.LOGREC, an SVC dump is scheduled, and queue manager termination is requested. The termination reason code reflects the function for which retry was unsuccessfully attempted.

System programmer response

This is a secondary error. Obtain a copy of SYS1.LOGREC and the SVC dump for this error and for the original problem that resulted in the retry attempt. Examine the SYS1.LOGREC information and the dumps from both the original and the secondary error to determine if the recovery parameter area was damaged or if retry incorrectly restored registers for the mainline module.

Restart the queue manager.

00D90002

The recovery manager startup notification routine received an error return code from the recovery log manager when attempting to read a recovery manager status table (RMST) record from the bootstrap data set (BSDS) in one of the following cases:

- When reading the record containing the RMST header. The first copy was successfully read, but the second copy could not be found.
- When reading records containing the RMST entries. A *no record found* condition was encountered before all entries were read.
- When reading either a header record or an entry record. The record exceeded its expected length.

This is an IBM MQ error.

System action

The recovery manager has no functional recovery routine (FRR) in place when this error occurs. It relies on its invoker, the facility startup function, to perform SYS1.LOGREC recording and to request a dump. The queue manager terminates with a X'00E80100' reason code.

System programmer response

The queue manager determined that the BSDS that it was reading has been corrupted. If you are running in a dual BSDS environment, determine which BSDS is corrupt, and follow the procedures described in Recovering the BSDS to recover it from the valid BSDS.

Similarly, if you are running in a single BSDS environment, refer to Recovering the BSDS, which describes the procedures needed to recover your BSDS from an archived BSDS.

00D92001

The checkpoint/restart serial controller FRR invoked queue manager termination, because an unrecoverable error was detected while processing a request.

This is a queue manager termination reason code.

System action

Queue manager termination is initiated. Standard diagnostic information is recorded in SYS1.LOGREC, and an SVC dump is requested for the associated error.

System programmer response

Obtain a copy of the SYS1.LOGREC and the SVC dump for the original error, and follow the instructions associated with it.

Restart the queue manager.

00D92003

The restart request servicer FRR invoked queue manager termination, because an unrecoverable error was detected while processing a restart request.

This is a queue manager termination reason code.

System action

Standard diagnostic information is recorded in SYS1.LOGREC, and an SVC dump is requested for the original error before queue manager termination is initiated.

System programmer response

Obtain a copy of SYS1.LOGREC and the SVC dump for the original error and follow the instructions associated with it.

Restart the queue manager.

00D92004

The shutdown checkpoint controller FRR invoked queue manager termination, because an unrecoverable error was detected while processing a shutdown checkpoint request.

This is a queue manager termination reason code.

System action

Standard diagnostic information is recorded in SYS1.LOGREC, and an SVC dump is requested for the original error before queue manager termination is initiated.

System programmer response

Obtain a copy of SYS1.LOGREC and the SVC dump for the original error and follow the instructions associated with it.

Restart the queue manager.

00D92011

An internal error has occurred.

System action

The checkpoint process will end abnormally to prevent a damaged URE from being written out to the log, and the queue manager will be terminated. This is to prevent the loss or incorrect processing of an IBM MQ unit of recovery (UR). Restart will use the previous checkpoint and apply all the IBM MQ log records up to the point of the problem. Standard diagnostic information is recorded in SYS1.LOGREC, and an SVC dump is scheduled.

System programmer response

Restart the queue manager.

Collect the items listed in "Diagnostics" on page 1043 and contact IBM Support.

00D92012

An internal error has occurred.

System action

The checkpoint process will end abnormally to prevent a damaged RURE from being written out to the log, and the queue manager will be terminated. This is to prevent the loss or incorrect processing of an IBM MQ unit of recovery. Restart will use the previous checkpoint and apply all the IBM MQ log records up to the point of the problem. Standard diagnostic information is recorded in SYS1.LOGREC, and an SVC dump is scheduled.

System programmer response

Restart the queue manager.

Collect the items listed in "Diagnostics" on page 1043 and contact IBM Support.

00D92021

An internal error has occurred.

System action

The restart processing ends abnormally, which terminates the queue manager. This is to prevent the loss or incorrect processing of an IBM MQ unit of recovery.

System programmer response

Do not attempt to restart the queue manager until the error is resolved.

The log has become corrupted. If you are running with dual logging, try to start the queue manager from the undamaged log.

If you are unable to do achieve this, use the following procedure (you will lose all updates since your last back up):

1. Restore your page sets from the last set of full backups. The queue manager must have been shut down cleanly before taking the backup copies of the page sets.
2. Clear the logs by following the procedure detailed in Resetting the queue manager's log.

See Using the CSQUTIL utility for IBM MQ for z/OS for information about restarting the queue manager from one log when using dual logging. If you are unable to resolve the problem, contact IBM Support.

00D92022

An internal error has occurred.

System action

The restart processing ends abnormally, which terminates the queue manager. This is to prevent the loss or incorrect processing of an IBM MQ unit of recovery.

System programmer response

Do not attempt to restart the queue manager until the error is resolved.

The log has become corrupted. If you are running with dual logging, try to start the queue manager from the undamaged log.

If you are unable to do achieve this, use the following procedure (you will lose all updates since your last back up):

1. Restore your page sets from the last set of full backups. The queue manager must have been shut down cleanly before taking the backup copies of the page sets.
2. Clear the logs by following the procedure detailed in Resetting the queue manager's log.

See Using the CSQUTIL utility for IBM MQ for z/OS for information about restarting the queue manager from one log when using dual logging. If you are unable to resolve the problem, contact IBM Support.

00D92023

During queue manager restart in 6 byte log RBA mode, a log record has been encountered that is written with an 8 byte log RBA.

System action

The restart processing ends abnormally, which terminates the queue manager. This is to prevent the loss, or incorrect processing, of an IBM MQ unit of recovery.

System programmer response

Do not attempt to restart the queue manager until the error is resolved.

The queue manager might have been started with an incorrect log or BSDS. Ensure that the queue manager is started with the correct log and BSDS data sets. If this was not the cause of the problem, the log or BSDS has become corrupted.

To recover from a corrupted log or BSDS, if you are running with dual logging or dual BSDS, try to start the queue manager from the undamaged log.

If you are unable to do achieve this, use the following procedure. Note, that by carrying out this procedure, you will lose all updates since your last back up:

1. Restore your page sets from the last set of full backups. The queue manager must have been shut down cleanly before taking the backup copies of the page sets.
2. Clear the logs by following the procedure detailed in Resetting the queue manager's log.

If necessary, when clearing the logs ensure that you convert the BSDS to the previous version, using the BSDS conversion utility CSQJUCNV.

See Using the CSQUTIL utility for IBM MQ for z/OS. If you are unable to resolve the problem, contact IBM Support.

00D93001

The commit/backout FRR invoked queue manager termination, because an unrecoverable error was detected during 'must-complete' processing for phase 2 of a commit-UR request.

This is a queue manager termination reason code.

System action

Standard diagnostic information is recorded in SYS1.LOGREC, and an SVC dump is requested for the original error before queue manager termination is initiated.

System programmer response

Obtain a copy of SYS1.LOGREC and the SVC dump for the original error and follow the instructions associated with it.

Restart the queue manager.

00D93011

A subcomponent of IBM MQ invoked commit when the agent state was invalid for commit-UR invocation. Commit-UR was requested for an agent that was modifying data. Either commit-UR or backout-UR was already in process, or the recovery structure (URE) was damaged.

System action

Abnormal termination of the agent results, including backing out (backout-UR) of its activity to the previous point of consistency. This releases all locks held by the agent for its resources.

Standard diagnostic information is recorded in SYS1.LOGREC, and an SVC dump is scheduled. Additional information, identified in the SDWA variable recording area (VRA) by reason code X'00D9CCCC', is added to the VRA.

If the agent was in a 'must-complete' state (in-commit or in-backout), the queue manager is also terminated with reason code X'00D93001'. When the queue manager is next restarted, recoverable activity for this agent (such as an ensure-backout or ensure-commit UR) is handled to complete the commit or backout process.

System programmer response

This is an IBM MQ error. Examine the SYS1.LOGREC data and the dump to establish whether either commit-UR was invoked incorrectly or the control structure that reflects the state was damaged.

00D93012

A subcomponent of IBM MQ invoked commit when the agent state was invalid for commit-UR invocation. Commit-UR was invoked for an agent that was only retrieving data. Either commit-UR or backout-UR was already in process, or the ACE progress state field was damaged.

System action

Abnormal termination of the agent results, including backing out (backout-UR) of its activity to the previous point of consistency. This releases all locks held by the agent for its resources.

Standard diagnostic information is recorded in SYS1.LOGREC, and an SVC dump is scheduled. Additional information, identified in the SDWA variable recording area (VRA) by reason code X'00D9CCCC', is added to the SDWA VRA.

System programmer response

This is an IBM MQ error. Examine the SYS1.LOGREC data and the dump to establish whether either commit-UR was invoked incorrectly or the control structure was damaged.

00D93100

This reason code indicates that an IBM MQ allied agent does not need to participate in the Phase-2 (Continue Commit) call, because all required work has been accomplished during the Phase-1 (Prepare) call.

This reason code is generated by the recovery manager when it is determined that an IBM MQ allied agent has not updated any IBM MQ resource since its last commit processing occurred.

System action

The 'yes' vote is registered with the commit coordinator.

System programmer response

None should be required because this is not an error reason code. This reason code is used for communication between components of IBM MQ.

00D94001

The commit/backout FRR invoked queue manager termination, because an unrecoverable error was detected during 'must-complete' processing for a backout-UR request.

This is a queue manager termination reason code.

System action

Standard diagnostic information is recorded in SYS1.LOGREC, and an SVC dump is requested for the original error before queue manager termination is initiated.

System programmer response

Obtain a copy of SYS1.LOGREC and the SVC dump for the original error and follow the instructions associated with it.

Restart the queue manager.

00D94011

A subcomponent of IBM MQ invoked backout at a point when the agent state is invalid for invoking the function that backs out units of recovery. Either backout-UR or commit-UR phase-2 was already in process, or the agent structure was damaged.

System action

Abnormal termination of the agent results and, because the agent is in a 'must-complete' state, the queue manager is terminated with reason code X'00D94001'. When the queue manager is restarted, recoverable activity for this agent is handled to complete the commit or backout process.

Standard diagnostic information is recorded in SYS1.LOGREC, and an SVC dump is scheduled. Additional information, identified in the SDWA variable recording area (VRA) by reason code X'00D9AAAA', is added to the SDWA VRA.

System programmer response

This is an IBM MQ error. Examine the SYS1.LOGREC data and the dump to establish whether commit-UR was invoked incorrectly or the control structure was damaged.

00D94012

During backout, the end of the log was read before all the expected log ranges had been processed. The error is accompanied by an abnormal termination with reason code X'00D94001'.

This could be because the queue manager has been started with a system parameter load module that specifies OFFLOAD=NO rather than OFFLOAD=YES.

System action

The agent is abnormally terminated with completion code X'5C6'. Because the agent is in a must-complete state, the queue manager is terminated with reason code X'00D94001' and message CSQV086E.

Standard diagnostic information is recorded in SYS1.LOGREC. and an SVC dump is requested.

System programmer response

Run the print log map utility to print the content of both BSDSs. Obtain a copy of the SYS1.LOGREC and the SVC dump for the original error. At the time of the error, registers 3 and 4 contain the 8-byte relative byte address (RBA) of the beginning of this unit of recovery. IBM MQ must read the log back to this point to complete the backout of this unit of recovery.

To restart the queue manager, you must add the missing archive log data sets back to the BSDS with the change log inventory utility, and increase the MAXARCH parameter in the CSQ6LOGP macro (the system parameter module log initialization macro) to complete the backout.

If the missing archive log is not available, or if archiving was not active, the queue manager cannot be restarted unless the log data sets and page sets are all reinitialized or restored from backup copies. Data will be lost as a result of this recovery action.

00D95001

The recovery manager's common FRR invoked queue manager termination, because an unrecoverable error was detected during checkpoint processing.

This is a queue manager termination reason code.

System action

Standard diagnostic information is recorded in SYS1.LOGREC, and an SVC dump is requested for the original error before queue manager termination is initiated.

System programmer response

Obtain a copy of SYS1.LOGREC and the SVC dump for the original error and follow the instructions associated with it.

Restart the queue manager.

00D95011

The recovery manager checkpoint FRR invoked queue manager termination, because an unrecoverable error was detected while performing its checkpoint functions.

This is a queue manager termination reason code.

System action

Standard diagnostic information is recorded in SYS1.LOGREC, and an SVC dump is requested for the original error before queue manager termination is initiated.

System programmer response

Obtain a copy of the SYS1.LOGREC and the SVC dump for the original error and follow the instructions associated with it.

Restart the queue manager.

00D96001

The recovery manager's restart FRR invoked queue manager termination, because an unrecoverable error was detected during the restart processor processing.

This is a queue manager termination reason code.

System action

Standard diagnostic information is recorded in SYS1.LOGREC, and an SVC dump is requested for the original error before queue manager termination is initiated.

System programmer response

Obtain a copy of the SYS1.LOGREC and the SVC dump for the original error and follow the instructions associated with it.

Restart the queue manager.

00D96011

The restart participation FRR invoked queue manager termination, because an unrecoverable error was detected while processing log records during restart.

This is a queue manager termination reason code.

System action

Standard diagnostic information is recorded in SYS1.LOGREC, and an SVC dump is requested for the original error before queue manager termination is initiated.

System programmer response

Obtain a copy of the SYS1.LOGREC and the SVC dump for the original error and follow the instructions associated with it.

Restart the queue manager when the problem has been corrected.

00D96021

The queue manager was terminated during restart because an error occurred while attempting to read the log forward MODE(DIRECT). It is accompanied by a recovery log manager error X'5C6' with a reason code describing the specific error.

Each time a portion of the log is skipped, a 'read direct' is used to validate the beginning RBA of the portion that is read.

This is a queue manager termination reason code.

System action

Standard diagnostic information is recorded in SYS1.LOGREC, and an SVC dump is requested for the original error before queue manager termination is initiated.

System programmer response

Run the print log map utility to print the contents of both BSDSs. Obtain a copy of the SYS1.LOGREC and the SVC dump for the original error. Follow instructions for the accompanying recovery log manager error. If possible, remove the cause of original error and restart the queue manager. If you cannot correct the error, contact IBM Support.

00D96022

The restart FRR invoked abnormal termination, because, while reading the log forward during restart, the end-of-log was read before all recovery log scopes had been processed. It is followed by an abnormal termination with the same reason code (X'00D96022').

This is a queue manager termination reason code.

System action

Standard diagnostic information is recorded in SYS1.LOGREC, and an SVC dump is requested for the error before queue manager termination is initiated.

System programmer response

Run the print log map utility to print the contents of both BSDSs. Obtain a copy of the SYS1.LOGREC and the SVC dump for the original error. At the time of the error, registers 2 and 3 (as shown in the dump or in SYS1.LOGREC) contain the relative byte address (RBA) of the last log record that was read before end-of-log was encountered. Follow instructions for the accompanying recovery log manager error. If you cannot correct the error, contact IBM Support.

00D96031

The restart FRR invoked queue manager termination, because an error occurred while attempting to read the log backward MODE(DIRECT). It is accompanied by a recovery log manager error X'5C6' with a reason code describing the specific error.

Each time a portion of the log is skipped, a 'read direct' is used to validate the beginning RBA of the portion that is read.

This is a queue manager termination reason code.

System action

Standard diagnostic information is recorded in SYS1.LOGREC, and an SVC dump is requested for the original error before queue manager termination is initiated.

System programmer response

Run the print log map utility to print the contents of both BSDSs. Obtain a copy of the SYS1.LOGREC and the SVC dump for the original error. Follow instructions for the accompanying recovery log manager error. See the accompanying error reason code.

Restart the queue manager.

00D96032

During restart, the end of the log was read before all the expected log ranges had been processed. The error is accompanied by an abnormal termination with the same reason code (X'00D96032').

This is a queue manager termination reason code.

System action

Standard diagnostic information is recorded in SYS1.LOGREC. An SVC dump is requested. The queue manager is terminated with message CSQV086E.

System programmer response

Run the print log map utility to print the contents of both BSDSs. See Finding out what the BSDS contains for more information.

Obtain a copy of the SYS1.LOGREC and the SVC dump for the original error. At the time of the error, registers 2 and 3 contain the relative byte address (RBA) of the last log record that was read before end-of-log was encountered. Determine where the log went.

00D97001

The agent concerned was canceled while waiting for the RECOVER-UR service to complete.

System action

The RECOVER-UR function is completed. Abnormal termination of the requesting agent occurs. Standard diagnostic information is recorded in SYS1.LOGREC, and an SVC dump is requested.

The condition that caused cancellation of the agent was installation initiated (for example, a *forced* termination of the queue manager).

00D97011

The queue manager was terminated during RECOVER-UR because an unrecoverable error was detected during RECOVER-UR (CSQRRUPR) recovery processing.

This is a queue manager termination reason code.

System action

Standard diagnostic information is recorded in SYS1.LOGREC, and an SVC dump is requested. queue manager terminates with message CSQV086E and return code X'00D97011'.

System programmer response

Determine the original error. If the error is log-related, see Active log problems before restarting the queue manager.

00D97012

The RECOVER-UR request servicer FRR invoked queue manager termination, because an unrecoverable error was detected while attempting to recover a unit of recovery.

This is a queue manager termination reason code.

System action

Standard diagnostic information is recorded in SYS1.LOGREC, and an SVC dump is requested for the original error before queue manager termination is initiated.

System programmer response

Obtain a copy of the SYS1.LOGREC and the SVC dump for the original error and follow the instructions associated with it.

Restart the queue manager.

00D97021

The RECOVER-UR FRR invoked queue manager termination, because an error occurred while attempting to read the log MODE(DIRECT) during forward processing. It is accompanied by a recovery log manager error X'5C6' with a reason code describing the specific error.

Each time a portion of the log is skipped, a 'read direct' is used to validate the beginning RBA of the portion that is read.

This is a queue manager termination reason code.

System action

Standard diagnostic information is recorded in SYS1.LOGREC, and an SVC dump is requested for the original error before queue manager termination is initiated.

System programmer response

Run the print log map utility to print the contents of both BSDSs. Obtain a copy of the SYS1.LOGREC and the SVC dump for the original error. Follow instructions for the accompanying recovery log manager error. See the accompanying error reason code.

Restart the queue manager.

00D97022

The RECOVER-UR invoked abnormal termination because end-of-log was reached before all ranges had been processed for forward recovery. This error is accompanied by an abnormal termination with the same reason code (X'00D97022').

This is a queue manager termination reason code.

System action

Standard diagnostic information is recorded in SYS1.LOGREC, and an SVC dump is requested for the original error before queue manager termination is initiated.

System programmer response

Run the print log map utility to print the contents of both BSDSs. Obtain a copy of the SYS1.LOGREC and the SVC dump for the original error. At the time of the error, registers 2 and 3 contain the relative byte address (RBA) of the last log record that was read before end-of-log was encountered. Follow instructions for the accompanying recovery log manager error.

Restart the queue manager.

00D97031

The RECOVER-UR FRR invoked queue manager termination, because an error occurred during an attempt to read the log MODE(DIRECT) while reading the log backward. It is accompanied by a recovery log manager error X'5C6' with a reason code describing the specific error.

Each time a portion of the log is skipped, a 'read direct' is used to validate the begin-scope RBA of the portion that is read.

This is a queue manager termination reason code.

System action

Standard diagnostic information is recorded in SYS1.LOGREC, and an SVC dump is requested for the original error before queue manager termination is initiated.

System programmer response

Run the print log map utility to print the contents of both BSDSs. Obtain a copy of the SYS1.LOGREC and the SVC dump for the original error. See the accompanying error reason code. Follow instructions for the accompanying recovery log manager error.

Restart the queue manager.

00D97032

The RECOVER-UR invoked abnormal termination because end-of-log was reached before all ranges had been processed for backward recovery. This error is accompanied by an abnormal termination with the same reason code (X'00D97032').

This is a queue manager termination reason code.

System action

Standard diagnostic information is recorded in SYS1.LOGREC, and an SVC dump is requested for the original error before queue manager termination is initiated.

System programmer response

Run the print log map utility to print the contents of both BSDSs. Obtain a copy of the SYS1.LOGREC and the SVC dump for the original error. At the time of the error, registers 2 and 3 contain the relative byte address (RBA) of the last log record that was read before end-of-log was encountered. Follow instructions for the accompanying recovery log manager error.

Restart the queue manager.

00D98001

The recovery manager's common FRR invoked queue manager termination, because an unrecoverable error was detected during indoubt-UR processing.

This is a queue manager termination reason code.

System action

Standard diagnostic information is recorded in SYS1.LOGREC, and an SVC dump is requested for the original error before queue manager termination is initiated.

System programmer response

Obtain a copy of the SYS1.LOGREC and the SVC dump for the original error and follow the instructions associated with it.

Restart the queue manager.

00D98011

The FRR for the resolved-indoubt-UR request servicer invoked queue manager termination, because an unrecoverable error was detected processing a request.

This is a queue manager termination reason code.

System action

Standard diagnostic information is recorded in SYS1.LOGREC, and an SVC dump is requested for the original error before queue manager termination is initiated.

System programmer response

Obtain a copy of the SYS1.LOGREC and the SVC dump for the original error. See the accompanying error reason code.

Restart the queue manager.

00D98021

The resolved indoubt FRR invoked queue manager termination because of an error while attempting to read the log MODE(DIRECT) during forward recovery. It is accompanied by a recovery log manager error X'5C6' with a reason code describing the specific error.

Each time a portion of the log is skipped, a 'read direct' is used to validate the beginning RBA of the portion that is read.

This is a queue manager termination reason code.

System action

Standard diagnostic information is recorded in SYS1.LOGREC, and an SVC dump is requested for the original error before queue manager termination is initiated.

System programmer response

Run the print log map utility to print the contents of both BSDSs. Obtain a copy of the SYS1.LOGREC and the SVC dump for the original error. See the accompanying error reason code. Follow instructions for the accompanying recovery log manager error.

Restart the queue manager.

00D98022

Resolved indoubt invoked abnormal termination when end-of-log was reached before all ranges had been processed for forward recovery. This error is accompanied by abnormal termination with the same reason code (X'00D98022').

This is a queue manager termination reason code.

System action

Standard diagnostic information is recorded in SYS1.LOGREC, and an SVC dump is requested for the original error before queue manager termination is initiated.

System programmer response

Run the print log map utility to print the contents of both BSDSs. Obtain a copy of the SYS1.LOGREC and the SVC dump for the original error. At the time of the error, registers 2 and 3 contain the relative byte address (RBA) of the last log record that was read before end-of-log was encountered. Follow instructions for the accompanying recovery log manager error.

Restart the queue manager.

00D98031

The resolved indoubt FRR invoked queue manager termination, because an error occurred during an attempt to read the log MODE(DIRECT) while reading the log backward. It is accompanied by a recovery log manager error X'5C6' with a reason code describing the specific error.

Each time a portion of the log is skipped, a 'read direct' is used to validate the begin-scope RBA of the portion that is read.

This is a queue manager termination reason code.

System action

Standard diagnostic information is recorded in SYS1.LOGREC, and an SVC dump is requested for the original error before queue manager termination is initiated.

System programmer response

Run the print log map utility to print the contents of both BSDSs. Obtain a copy of the SYS1.LOGREC and the SVC dump for the original error. See the accompanying error reason code. Follow instructions for the accompanying recovery log manager error.

Restart the queue manager.

00D98032

The resolved indoubt FRR invoked abnormal termination when end-of-log was reached before all ranges had been processed for backward recovery. This error is accompanied by abnormal termination with the same reason code (X'00D98032').

This is a queue manager termination reason code.

System action

Standard diagnostic information is recorded in SYS1.LOGREC, and an SVC dump is requested for the original error before queue manager termination is initiated.

System programmer response

Run the print log map utility to print the contents of both BSDSs. Obtain a copy of the SYS1.LOGREC and the SVC dump for the original error. At the time of the error, registers 2 and 3 contain the relative byte address (RBA) of the last log record that was read before end-of-log was encountered. Follow instructions for the accompanying recovery log manager error.

Restart the queue manager.

00D99001

The checkpoint RBA in the conditional restart control record, which is deduced from the end RBA or LRSN value that was specified, is not available. This is probably because the log data sets available for use at restart do not include that end RBA or LRSN.

System action

The queue manager terminates.

System programmer response

See message CSQR015E.

00D99104

Queue manager restart detected that backward migration of messages was required. For backward migration to be possible, there must be no uncommitted units of recovery present at the end of restart. During restart, however, a decision was made not to force commit a detected indoubt unit of work. The decision is based on the response to message CSQR021D, or by the presence of a service parm which prevents the CSQR021D WTOR from being issued.

System action

Queue manager restart is terminated.

System programmer response

Either restart the queue manager with a higher level of code so that backward migration is not required, or, allow indoubt units of work to be force committed during restart.

00D9AAAA

This reason code identifies additional data stored in the system diagnostic work area (SDWA) variable recording area (VRA) following an error during backout-UR.

System action

Data is stored in the field indicated by VRA key 38 following the EBCDIC string 'RMC-COMMIT/BACKOUT'. This information is useful for IBM service personnel.

System programmer response

Quote this code, and the contents of the VRA field indicated by key 38 when contacting IBM Support.

00D9BBBB

This reason code identifies additional data stored in the system diagnostic work area (SDWA) variable recording area (VRA) following an error during begin-UR.

System action

Data is stored in the field indicated by VRA key 38. This information is useful for IBM service personnel.

System programmer response

Quote this code, and the contents of the VRA field indicated by key 38 when contacting IBM Support.

00D9CCCC

This reason code identifies additional data stored in the system diagnostic work area (SDWA) variable recording area (VRA) following an error during commit-UR.

System action

Data is stored in the field indicated by VRA key 38 following the EBCDIC string 'RMC-COMMIT/ABORT'. This information is useful for IBM service personnel.

System programmer response

Quote this code, and the contents of the VRA field indicated by key 38 when contacting IBM Support.

00D9EEEE

This reason code identifies additional data stored in the system diagnostic work area (SDWA) variable recording area (VRA) following an error during end-UR.

System action

Data is stored in the field indicated by VRA key 38. This information is useful for IBM service personnel.

System programmer response

Quote this code, and the contents of the VRA field indicated by key 38 when contacting IBM Support.

Related tasks

Collecting troubleshooting information

z/OS *Storage manager codes (X'E2')*

If a storage manager reason code occurs that is not listed here, an internal error has occurred. Collect the following diagnostic items and contact IBM Support.

Diagnostics

- A description of the actions that led to the error or, if applicable, either a listing of the application program or the input string to a utility program that was being run at the time of the error.
- The console output for the period leading up to the error.
- The queue manager job log.
- The system dump resulting from the error.
- A printout of SYS1.LOGREC.
- If you are using CICS , the CICS transaction dump output.
- Appropriate IBM MQ, z/OS, Db2, CICS, and IMS service levels.

00E20001, 00E20002

An internal error has occurred.

System action

The invoker is abnormally terminated. Diagnostic information is recorded in SYS1.LOGREC, and a dump is requested.

System programmer response

Collect the items listed in “Diagnostics” on page 1056 and contact IBM Support.

00E20003

A request for storage indicated that sufficient storage in the private area was not available.

System action

The invoker is abnormally terminated. Diagnostic information is recorded in SYS1.LOGREC, and a dump is requested.

System programmer response

Increase region size.

If you are unable to solve the problem by increasing the region size, collect the items listed in “Diagnostics” on page 1056 and contact IBM Support.

00E20004

A request for storage indicated that sufficient storage was not available because of pool size limits.

System action

The invoker is abnormally terminated. Diagnostic information is recorded in SYS1.LOGREC, and a dump is requested.

System programmer response

Increase pool sizes.

If you are unable to solve the problem by increasing the pool sizes, collect the items listed in “Diagnostics” on page 1056 and contact IBM Support.

00E20005, 00E20006, 00E20007, 00E20008, 00E20009

An internal error has occurred.

System action

The invoker is abnormally terminated. Diagnostic information is recorded in SYS1.LOGREC, and a dump is requested.

System programmer response

Collect the items listed in “Diagnostics” on page 1056 and contact IBM Support.

00E2000A

A request to get storage was unsuccessful.

System action

The invoker is abnormally terminated. Diagnostic information is recorded in SYS1.LOGREC, and a dump is requested.

System programmer response

Increase the region size.

If increasing the region size does not help you solve the problem, collect the items listed in “Diagnostics” on page 1056 and contact IBM Support.

00E2000B

A request to get storage was unsuccessful.

System action

The invoker is abnormally terminated. Diagnostic information is recorded in SYS1.LOGREC, and a dump is requested.

System programmer response

Increase region size.

If increasing the region size does not help you solve the problem, collect the items listed in “Diagnostics” on page 1056 and contact IBM Support.

00E2000C

A request for storage indicated that sufficient storage was not available because of pool size limits.

System action

The invoker is abnormally terminated. Diagnostic information is recorded in SYS1.LOGREC, and a dump is requested.

System programmer response

Increase pool sizes.

If increasing the pool size does not help you solve the problem, collect the items listed in “Diagnostics” on page 1056 and contact IBM Support.

00E2000D, 00E2000E

An internal error has occurred.

System action

The invoker is abnormally terminated. Diagnostic information is recorded in SYS1.LOGREC, and a dump is requested.

System programmer response

The most likely cause of the problem is a storage overlay or an invalid storage request from a queue manager component. A product other than MQ could cause the storage overlay problem.

Collect the items listed in “Diagnostics” on page 1056 and contact IBM Support.

00E2000F, 00E20010, 00E20011, 00E20012

An internal error has occurred.

System action

The invoker is abnormally terminated. Diagnostic information is recorded in SYS1.LOGREC, and a dump is requested.

System programmer response

Collect the items listed in “Diagnostics” on page 1056 and contact IBM Support.

00E20013

A request to get storage was unsuccessful.

System action

The invoker is abnormally terminated. Diagnostic information is recorded in SYS1.LOGREC, and a dump is requested.

System programmer response

Increase region size.

If increasing the region size does not help you to solve the problem, collect the items listed in “Diagnostics” on page 1056 and contact IBM Support.

00E20014

An internal error has occurred.

System action

The invoker is abnormally terminated. Diagnostic information is recorded in SYS1.LOGREC, and a dump is requested.

System programmer response

Collect the items listed in “Diagnostics” on page 1056 and contact IBM Support.

00E20015

A request for storage indicated that 8K bytes of private area storage in subpool 229 was not available.

System action

The invoker is abnormally terminated. Diagnostic information is recorded in SYS1.LOGREC, and a dump is requested.

System programmer response

There is probably a shortage of private area storage in the address space in which the problem occurred. Increase maximum private storage.

If increasing the maximum private storage does not solve the problem, collect the items listed in “Diagnostics” on page 1056 and contact IBM Support.

00E20016

A request for storage indicated that sufficient storage in subpool 229 was not available.

System action

The invoker is abnormally terminated. Diagnostic information is recorded in SYS1.LOGREC, and a dump is requested.

System programmer response

Increase region size.

If increasing the region size does not help you resolve the problem, collect the items listed in “Diagnostics” on page 1056 and contact IBM Support.

00E20017, 00E20018, 00E20019

An internal error has occurred.

System action

The invoker is abnormally terminated. Diagnostic information is recorded in SYS1.LOGREC, and a dump is requested.

System programmer response

Collect the items listed in “Diagnostics” on page 1056 and contact IBM Support.

00E2001A

An error has occurred with the z/OS ESTAE.

System action

The invoker is abnormally terminated. Diagnostic information is recorded in SYS1.LOGREC, and a dump is requested. Register 15 contains the return code from the z/OS ESTAE.

System programmer response

Collect the items listed in “Diagnostics” on page 1056 and contact IBM Support.

00E2001B

The 'setlock obtain' function issued a nonzero return code.

System action

The invoker is abnormally terminated. Diagnostic information is recorded in SYS1.LOGREC, and a dump is requested.

System programmer response

Collect the items listed in “Diagnostics” on page 1056 and contact IBM Support.

00E2001D, 00E2001E

An internal error has occurred.

System action

The invoker is abnormally terminated. Diagnostic information is recorded in SYS1.LOGREC, and a dump is requested.

System programmer response

Collect the items listed in “Diagnostics” on page 1056 and contact IBM Support.

00E2001F

There was insufficient storage in the common service area (CSA) to satisfy a request for storage.

System action

The invoker is abnormally terminated. Diagnostic information is recorded in SYS1.LOGREC, and a dump is requested.

System programmer response

Run the monitoring tools available at your installation to review your CSA usage.

Increase the CSA size.

If increasing the CSA size does not solve the problem, collect the items listed in “Diagnostics” on page 1056 and contact IBM Support.

00E20020

There was insufficient storage in the private area to satisfy a request for storage.

System action

The invoker is abnormally terminated. Diagnostic information is recorded in SYS1.LOGREC, and a dump is requested.

System programmer response

Increase region size.

If increasing the region size does not solve the problem, collect the items listed in “Diagnostics” on page 1056 and contact IBM Support.

00E20021

There was insufficient storage in the common service area (CSA) to satisfy a request for storage.

System action

The invoker is abnormally terminated. Diagnostic information is recorded in SYS1.LOGREC, and a dump is requested.

System programmer response

Run the monitoring tools available at your installation to review your CSA usage.

Increase the CSA size.

If increasing the size of the CSA does not solve the problem, collect the items listed in “Diagnostics” on page 1056 and contact IBM Support.

00E20022

There was insufficient storage in the common service area (CSA) to satisfy a request for storage.

System action

The invoker is abnormally terminated. Diagnostic information is recorded in SYS1.LOGREC, and a dump is requested.

System programmer response

Run the monitoring tools available at your installation to review your CSA usage.

Increase the CSA size.

If increasing the size of the CSA does not solve the problem, collect the items listed in “Diagnostics” on page 1056 and contact IBM Support.

00E20023

There was insufficient storage in the private area was to satisfy a request for storage.

System action

The invoker is abnormally terminated. Diagnostic information is recorded in SYS1.LOGREC, and a dump is requested.

System programmer response

Increase region size.

If increasing the region size does not solve the problem, collect the items listed in “Diagnostics” on page 1056 and contact IBM Support.

00E20024

There was insufficient storage in the common service area (CSA) to satisfy a request for storage.

System action

The invoker is abnormally terminated. Diagnostic information is recorded in SYS1.LOGREC, and a dump is requested.

System programmer response

Run the monitoring tools available at your installation to review your CSA usage.

Increase the CSA size.

If increasing the CSA size does not solve the problem, collect the items listed in “Diagnostics” on page 1056 and contact IBM Support.

00E20025

There was insufficient storage in the common service area (CSA) to satisfy a request for storage.

System action

The invoker is abnormally terminated. Diagnostic information is recorded in SYS1.LOGREC, and a dump is requested.

System programmer response

Run the monitoring tools available at your installation to review your CSA usage.

Increase the CSA size.

If increasing the CSA size does not solve the problem, collect the items listed in “Diagnostics” on page 1056 and contact IBM Support.

00E20026

A request for storage indicated that 4K bytes of private area storage in subpool 229 was not available.

System action

The invoker is abnormally terminated. Diagnostic information is recorded in SYS1.LOGREC, and a dump is requested.

System programmer response

There is probably a shortage of private area storage in the address space in which the problem occurred. Increase region size.

If increasing the region size does not solve the problem, collect the items listed in “Diagnostics” on page 1056 and contact IBM Support.

00E20027, 00E20028, 00E20029, 00E2002A

An internal error has occurred.

System action

The invoker is abnormally terminated. Diagnostic information is recorded in SYS1.LOGREC, and a dump is requested.

System programmer response

Collect the items listed in "Diagnostics" on page 1056 and contact IBM Support.

00E2002B

This reason code code is used to force percolation when an error is encountered while in storage manager code and the storage manager has been called recursively.

System programmer response

Refer to the originating error code.

00E20042, 00E20043, 00E20044, 00E20045

An internal error has occurred.

System action

The invoker is abnormally terminated. Diagnostic information is recorded in SYS1.LOGREC, and a dump is requested.

System programmer response

Collect the items listed in "Diagnostics" on page 1056 and contact IBM Support.

00E20046

There was insufficient storage in a 64-bit storage pool to satisfy a request.

System action

The invoker is abnormally terminated. Diagnostic information is recorded in SYS1.LOGREC, and a dump is requested.

System programmer response

Increase the MEMLIM for the queue manager and restart it. If the problem persists collect the items listed in "Diagnostics" on page 1056 and contact IBM Support.

00E20047

An internal error has occurred.

System action

The invoker is abnormally terminated. Diagnostic information is recorded in SYS1.LOGREC, and a dump is requested.

System programmer response

Collect the items listed in "Diagnostics" on page 1056 and contact IBM Support.

Related tasks

Collecting troubleshooting information

z/OS *Timer services codes (X'E3')*

00E30001

An internal error has occurred.

System programmer response

Collect the system dump, any trace information gathered and the related SYS1.LOGREC entries, and contact IBM Support.

00E30002

This reason code was issued because an attempt to call the z/OS macro STIMERM was unsuccessful. The return code from STIMERM is in register 9.

System programmer response

Analyze the system dump, correct the problem from the information contained in the dump, and restart the queue manager.

See *z/OS MVS Programming: Assembler Services Reference* Volume 2 For information about the STIMERM macro.

Related tasks

Collecting troubleshooting information

z/OS *Agent services codes (X'E5')*

If an agent services reason code occurs that is not listed here, an internal error has occurred. Collect the following diagnostic items and contact IBM Support.

Diagnostics

- A description of the actions that led to the error or, if applicable, either a listing of the application program or the input string to a utility program that was being run at the time of the error.
- The console output for the period leading up to the error.
- The queue manager job log.
- The system dump resulting from the error.
- A printout of SYS1.LOGREC.
- If you are using CICS , the CICS transaction dump output.
- Appropriate IBM MQ, z/OS, Db2, CICS, and IMS service levels.

00E50001, 00E50002

An internal error has occurred.

System action

The requesting execution unit is ended abnormally.

System programmer response

Collect the items listed in “Diagnostics” on page 1063 and contact IBM Support.

00E50004, 00E50005, 00E50006, 00E50007, 00E50008, 00E50009, 00E50012

An internal error has occurred.

System action

The requesting execution unit is ended abnormally. A record is written to SYS1.LOGREC and an SVC dump is requested.

System programmer response

Collect the items listed in “Diagnostics” on page 1063 and contact IBM Support.

00E50013

An MQ execution unit has been ended abnormally.

System action

The agent CANCEL processing continues.

System programmer response

This reason code might be issued as a result of any abnormal termination of a connected task, or a STOP QMGR MODE(FORCE) command. No further action is required in such cases.

If the error results in the termination of the queue manager, and you are unable to resolve the problem, collect the items listed in “Diagnostics” on page 1063 and contact IBM Support.

00E50014

An internal error has occurred.

System action

An entry is written to SYS1.LOGREC, and an SVC dump is requested.

System programmer response

Collect the items listed in “Diagnostics” on page 1063 and contact IBM Support.

00E50015

An internal error has occurred.

System action

The operation is retried once. If this is not successful, the queue manager is terminated with reason code X'00E50054'.

A SYS1.LOGREC entry and an SVC dump are taken.

System programmer response

Restart the queue manager if necessary.

Collect the items listed in “Diagnostics” on page 1063 and contact IBM Support.

00E50029

The agent services function which establishes the MQ tasking structure ends abnormally with this reason code following the detection of a load module which was loaded without the 31-bit addressing capability. This is preceded by message CSQV029E.

System action

Queue manager start-up is terminated.

System programmer response

See message CSQV029E.

00E50030, 00E50031, 00E50032, 00E50035, 00E50036

An internal error has occurred.

System action

The requesting execution unit is ended abnormally. The error is recorded on SYS1.LOGREC, and an SVC dump is requested.

System programmer response

Collect the items listed in “Diagnostics” on page 1063 and contact IBM Support.

00E50039

An attempt was made to free an ACE while it was still on a VLCAAACE chain.

System action

An SVC dump is requested specifying a completion code of X'5C6' and this reason code. Processing continues.

System programmer response

Collect the items listed in “Diagnostics” on page 1063 and contact IBM Support.

00E50040

Queue manager termination was invoked following an unrecoverable error while processing a terminate allied agent request at the *thread*, or *identify* level.

System action

The queue manager is terminated.

System programmer response

Restart the queue manager.

Scan the system log and the contents of SYS1.LOGREC for MQ errors occurring immediately before the system termination message CSQV086E. Follow the problem determination procedures for the specific errors. If you are unable to resolve the problem, collect the items listed in “Diagnostics” on page 1063 and contact IBM Support.

00E50041

Queue manager termination was invoked following an unrecoverable error while processing a terminate agent request.

System action

The queue manager is terminated.

System programmer response

Restart the queue manager.

Scan the system log and the contents of SYS1.LOGREC for MQ errors occurring immediately before the system termination message CSQV086E. Follow the problem determination procedures for the specific errors. If you are unable to resolve the problem, collect the items listed in “Diagnostics” on page 1063 and contact IBM Support.

00E50042, 00E50044

An internal error has occurred.

System action

The current execution unit is ended abnormally. A record is written to SYS1.LOGREC and an SVC dump is requested.

System programmer response

Collect the items listed in “Diagnostics” on page 1063 and contact IBM Support.

00E50045

Queue manager termination was invoked following an unrecoverable error while processing a create allied agent service request at the *thread*, or *identify* level.

System action

The queue manager is terminated.

System programmer response

Restart the queue manager.

Scan the system log and the contents of SYS1.LOGREC for MQ errors occurring immediately before the termination message CSQV086E. Follow the problem determination procedures for the specific errors. If you are unable to resolve the problem, collect the items listed in “Diagnostics” on page 1063 and contact IBM Support.

00E50046

Queue manager termination was invoked following an unrecoverable error while processing a create agent structure request.

System action

The queue manager is terminated.

System programmer response

Restart the queue manager.

Scan the system log and the contents of SYS1.LOGREC for MQ errors occurring immediately before the termination message CSQV086E. Follow the problem determination procedures for the specific errors. If you are unable to resolve the problem, collect the items listed in “Diagnostics” on page 1063 and contact IBM Support.

00E50047

An internal error has occurred.

System action

The queue manager is terminated.

System programmer response

Restart the queue manager.

Scan the system log and the contents of SYS1.LOGREC for MQ errors occurring immediately before the termination message CSQV086E. Follow the problem determination procedures for the specific errors. If you are unable to resolve the problem, collect the items listed in “Diagnostics” on page 1063 and contact IBM Support.

00E50050

An internal error has occurred.

System action

The requesting execution unit is ended abnormally.

An X'00E50054' recovery reason code is placed in the SDWACOMU field of the SDWA, indicating that synchronization services was responsible for queue manager termination.

System programmer response

Restart the queue manager.

Collect the items listed in “Diagnostics” on page 1063 and contact IBM Support.

00E50051

An internal error has occurred.

System action

The queue manager is ended abnormally with a X'5C6' completion code and this reason code.

An X'00E50054' recovery reason code is placed in the SDWACOMU field of the SDWA indicating that synchronization services was responsible for queue manager termination.

System programmer response

Restart the queue manager.

Collect the items listed in “Diagnostics” on page 1063 and contact IBM Support.

00E50052

The z/OS cross-memory lock (CML) could not be released.

System action

The queue manager is ended abnormally with a X'5C6' completion code and this reason code.

An X'00E50054' recovery reason code is placed in the SDWACOMU field of the SDWA indicating that synchronization services was responsible for queue manager termination.

A record is written to SYS1.LOGREC and an SVC dump is produced.

System programmer response

Restart the queue manager.

Collect the items listed in "Diagnostics" on page 1063 and contact IBM Support.

00E50054

The queue manager is ended abnormally by the synchronization services recovery routine when an unrecoverable error is encountered during recovery processing for the SUSPEND, CANCEL, RESUME, or SRB REDISPATCH functions. This is a queue manager termination reason code.

One of the following conditions was encountered during recovery processing for the requested function:

- Unable to complete resume processing for an SRB mode execution unit that was suspended at time of error
- Errors were encountered during primary recovery processing causing entry to the secondary recovery routine
- Recovery initiated retry to mainline suspend/resume code caused retry recursion entry into the functional recovery routine
- Unable to obtain or release the cross-memory lock (CML) of the queue manager address space either during mainline processing or during functional recovery processing (for example, reason code X'00E50052')

System action

The queue manager is terminated. This reason code is associated with a X'6C6' completion code indicating that synchronization services was responsible for termination.

System programmer response

Restart the queue manager.

Scan the system log and the contents of SYS1.LOGREC for IBM MQ errors occurring immediately before the system termination message CSQV086E. Follow the problem determination procedures for the specific errors. If you are unable to resolve the problem, collect the items listed in "Diagnostics" on page 1063 and contact IBM Support.

00E50055

The synchronization services functional recovery routine was unable to successfully complete resume processing for a suspended TCB mode execution unit. The resume processing was requested by the CANCEL or RESUME functions.

System action

Because the suspended TCB mode execution unit must not be permitted to remain in a suspended state, the recovery routine invokes the z/OS CALLRTM (TYPE=ABTERM) service to end the execution unit abnormally with a X'6C6' completion code. Depending upon which execution unit was terminated, the queue manager might be ended abnormally.

System programmer response

Restart the queue manager if necessary.

Scan the system log and the contents of SYS1.LOGREC for MQ errors occurring immediately before the end of the execution unit. Follow the problem determination procedures for the specific errors. If you

are unable to resolve the problem, collect the items listed in “Diagnostics” on page 1063 and contact IBM Support.

00E50059

An internal error has occurred.

System action

If the module detecting the error is CSQVSDC0, it will be retried once. If validation is unsuccessful, the queue manager is terminated abnormally with a X'00E50054' reason code.

A SYS1.LOGREC entry and an SVC dump are requested.

System programmer response

Restart the queue manager.

Collect the items listed in “Diagnostics” on page 1063 and contact IBM Support.

00E50062

An internal error has occurred.

System action

The allied task is ended abnormally.

System programmer response

Collect the items listed in “Diagnostics” on page 1063 and contact IBM Support.

00E50063

An internal error has occurred.

System action

The task is ended abnormally.

System programmer response

Collect the items listed in “Diagnostics” on page 1063 and contact IBM Support.

00E50065

An internal error has occurred.

System action

The execution unit is ended abnormally.

System programmer response

Collect the items listed in “Diagnostics” on page 1063 and contact IBM Support.

00E50069

This reason code is issued during recovery processing for the suspend function when executing in SRB mode under the recovery routine established by the z/OS SRBSTAT(SAVE) service. Because the recovery routine established by this service is the only routine in the FRR stack at the time of error, normal RTM percolation to the invoking resource manager recovery routine is not possible.

After recovery processing for the initial error has successfully completed, the RTM environment is exited through retry to a routine that restores the original FRR stack. This routine terminates abnormally with completion code X'5C6' and this reason code. This causes entry into the original recovery routine established during suspend initialization.

System action

After this is intercepted by the original suspend recovery routine, a SYS1.LOGREC entry and SVC dump are requested to document the original error. The original recovery reason code is placed in the SDWACOMU field of the SDWA indicating the actions performed during recovery processing of the

initial error. Control is then returned to the invoking resource manager's recovery routine through RTM percolation.

System programmer response

Because this is used only to permit the transfer of the initial recovery reason code to the invoking resource manager's recovery routine, no further recovery actions are required for this reason code. Diagnostic information for the initial error encountered can be obtained through the SYS1.LOGREC and SVC dump materials provided.

00E50070

To enable an internal task to terminate itself, the task has ended abnormally. This is not necessarily an error.

System action

The task is ended abnormally.

If the service task is ended abnormally with a completion code of X'6C6', no SVC dump is taken.

System programmer response

The error should be ignored if it happens in isolation, however, if it occurs in conjunction with other problems, these problems should be resolved.

If you are unable to resolve the problem, collect the items listed in “Diagnostics” on page 1063 and contact IBM Support.

00E50071

An internal error has occurred.

System action

The internal task is ended abnormally.

System programmer response

Collect the items listed in “Diagnostics” on page 1063 and contact IBM Support.

00E50072

An internal error has occurred.

System action

The queue manager is ended abnormally.

System programmer response

Restart the queue manager.

Collect the items listed in “Diagnostics” on page 1063 and contact IBM Support.

00E50073

An internal error has occurred.

System action

The current execution unit is ended abnormally. A record is written to SYS1.LOGREC, and an SVC dump is requested.

System programmer response

Collect the items listed in “Diagnostics” on page 1063 and contact IBM Support.

00E50074

This reason code is issued in response to a nonzero return code from ATTACH during an attempt to create an internal task.

System action

The ATTACH is retried. A record is written to SYS1.LOGREC, and an SVC dump is requested. If a problem occurs again, the queue manager is terminated.

System programmer response

Restart the queue manager if necessary.

Register 2, in the SDWA, contains the return code from the ATTACH request. If you are unable to resolve the problem, collect the items listed in "Diagnostics" on page 1063 and contact IBM Support.

00E50075, 00E50076, 00E50077, 00E50078

An internal error has occurred.

System action

The requesting execution unit is terminated. The queue manager might also be terminated. A record is written to SYS1.LOGREC, and an SVC dump is requested.

System programmer response

Restart the queue manager if necessary.

Collect the items listed in "Diagnostics" on page 1063 and contact IBM Support.

00E50079

An internal error has occurred. This can occur if the allied address space is undergoing termination.

System action

The requesting execution unit is ended abnormally. A record is written to SYS1.LOGREC, and an SVC dump is requested.

System programmer response

If you are unable to resolve the problem, collect the items listed in "Diagnostics" on page 1063 and contact IBM Support.

00E50080, 00E50081

An internal error has occurred.

System action

An SVC dump is requested specifying a completion code of X'5C6' and this reason code. No record is written to SYS1.LOGREC. Execution continues.

System programmer response

Collect the items listed in "Diagnostics" on page 1063 and contact IBM Support.

00E50094, 00E50095, 00E50096, 00E50097, 00E50100

An internal error has occurred.

System action

The requesting recovery routine is ended abnormally. A record is written to SYS1.LOGREC, and an SVC dump is requested.

System programmer response

Collect the items listed in "Diagnostics" on page 1063 and contact IBM Support.

00E50101

MQ was unable to establish an ESTAE.

System action

The error is passed on to a subsystem support subcomponent (SSS) ESTAE. Probably, the queue manager is ended abnormally. A record is written to SYS1.LOGREC, and an SVC dump is requested.

System programmer response

The inability to establish an ESTAE is normally due to insufficient free space in the local system queue area (LSQA) for an ESTAE control block (SCB). If necessary, increase the size of the queue manager address space.

Restart the queue manager.

Review the associated SVC dump for usage and free areas in the LSQA subpools belonging to the system services address space. If you are unable to solve the problem, collect the items listed in "Diagnostics" on page 1063 and contact IBM Support.

00E50102

An unrecoverable error occurred while canceling all active agents during processing of the STOP QMGR MODE(FORCE) command. This is a queue manager termination reason code.

System action

The queue manager is ended abnormally. A record is written to SYS1.LOGREC.

System programmer response

Restart the queue manager.

You might find the items listed in "Diagnostics" on page 1063 useful in resolving the problem. Review the SYS1.LOGREC entries for errors immediately preceding queue manager termination.

00E50500

A z/OS LOCAL or CML lock could not be obtained during queue manager abnormal termination processing.

System action

The execution unit is ended abnormally. The error is recorded on SYS1.LOGREC, and abnormal queue manager termination is completed under a different execution unit if possible.

System programmer response

Restart the queue manager if necessary.

You might find the items listed in "Diagnostics" on page 1063 useful in resolving the problem.

00E50501

A z/OS LOCAL or CML lock could not be released during queue manager abnormal termination processing.

System action

The execution unit is ended abnormally. The error is recorded on SYS1.LOGREC. Queue manager termination is completed under a different execution unit if possible.

System programmer response

Restart the queue manager.

You might find the items listed in "Diagnostics" on page 1063 useful in resolving the problem.

00E50502

A z/OS LOCAL lock could not be obtained during queue manager abnormal termination processing.

System action

The execution unit is ended abnormally. The error is recorded on SYS1.LOGREC, and abnormal queue manager termination is completed under a different execution unit if possible.

System programmer response

Restart the queue manager.

You might find the items listed in “Diagnostics” on page 1063 useful in resolving the problem.

00E50503

A z/OS LOCAL lock could not be released during queue manager abnormal termination processing.

System action

The execution unit is ended abnormally. The error is recorded on SYS1.LOGREC, and abnormal queue manager termination is completed under a different execution unit if possible.

System programmer response

Restart the queue manager.

You might find the items listed in “Diagnostics” on page 1063 useful in resolving the problem.

00E50504

This reason code is used to define the format of the information recorded in the SDWA variable recording area (VRA) by the queue manager termination processor. The code identifies additional information provided in the VRA for errors encountered in module CSQVATRM.

System action

Recording of the error encountered during queue manager termination continues.

System programmer response

None.

00E50505

This reason code is used to define the format of the information recorded in the SDWA variable recording area (VRA). The code identifies additional information provided in the VRA for errors encountered in module CSQVATR4.

System action

Recording of the error encountered during queue manager termination continues.

System programmer response

None.

00E50701

A problem occurred during Commit Phase-1. This is used to effect backout, deallocation, and end-UR processing.

System action

The queue manager is ended abnormally. A record is written to SYS1.LOGREC, and an SVC dump is requested.

System programmer response

Restart the queue manager.

If you are unable to resolve the problem, collect the items listed in “Diagnostics” on page 1063 and contact IBM Support.

00E50702

An error occurred while processing in SRB mode which could not be recovered.

SRB mode processing is often used internally by the queue manager to ensure data integrity and consistency of internal state. Where recovery is not possible, the queue manager is terminated with this reason code.

Most occurrences are due to internal errors which should be reported to IBM service for further investigation.

The error is also known to occur where log data sets have been reformatted, without reformatting the page sets (so they still contain active data). This situation can be resolved by user action.

System action

The queue manager is ended abnormally with this reason code. An SVC dump of the original error was requested by the recovery routine for CSQVEUS2 and a record written to SYS1.LOGREC.

System programmer response

Restart the queue manager.

Scan the SYS1.LOGREC entries looking for one or more MQ errors immediately prior to the queue manager termination. If you are unable to resolve the problem, collect the items listed in “Diagnostics” on page 1063 and contact IBM Support.

00E50703

This queue manager termination reason code is used following an error while attempting to resume a suspended execution unit. The successful completion of resume processing was 'indoubt'.

System action

The queue manager is ended abnormally. A record is written to SYS1.LOGREC, and an SVC dump is requested.

System programmer response

Restart the queue manager.

You might find the items listed in “Diagnostics” on page 1063 useful in resolving the problem.

00E50704

An internal error has occurred.

System action

The queue manager is terminated with this reason code. Additionally, if no SDWA was provided to the recovery routine, a dump is requested.

System programmer response

Restart the queue manager.

Scan the SYS1.LOGREC entries looking for one or more MQ errors immediately prior to the queue manager termination. If you are unable to resolve the problem, collect the items listed in “Diagnostics” on page 1063 and contact IBM Support.

00E50705

An internal error has occurred.

System action

The queue manager is ended abnormally.

System programmer response

Restart the queue manager.

Collect the items listed in “Diagnostics” on page 1063 and contact IBM Support.

00E50706

An internal error has occurred.

System action

The queue manager is terminated with this reason code. Additionally, if no SDWA was provided to the recovery routine, a dump is requested. A record is written to SYS1.LOGREC.

System programmer response

Restart the queue manager.

Scan the SYS1.LOGREC entries looking for one or more IBM MQ errors immediately prior to the queue manager termination. If you are unable to resolve the problem, collect the items listed in "Diagnostics" on page 1063 and contact IBM Support.

00E50707

An ESTAE could not be established.

System action

The queue manager is ended abnormally. A record is written to SYS1.LOGREC.

System programmer response

Review the usage and the free areas in the LSQA subpool of the queue manager address space. If necessary, increase the private area size of the address space.

Restart the queue manager.

If queue manager termination was requested by module CSQVRCT, a standard SVC dump was requested. If insufficient private storage is the cause of the problem, other MQ resource managers might have ended abnormally.

If you are unable to resolve the problem, collect the items listed in "Diagnostics" on page 1063 and contact IBM Support.

00E50708

An error occurred while connecting an allied agent to the queue manager address space. The connection must complete so that the allied agent can be terminated.

System action

The queue manager is terminated with this reason code. An SVC dump of the original error was requested and a record entered into SYS1.LOGREC.

System programmer response

Restart the queue manager.

Scan the SYS1.LOGREC entries looking for one or more IBM MQ errors immediately prior to the queue manager termination.

00E50709

An internal error has occurred.

System action

The queue manager is ended abnormally.

System programmer response

Restart the queue manager.

Scan the SYS1.LOGREC entries for one or more IBM MQ errors occurring immediately prior to the queue manager termination. If you are unable to resolve the problem, collect the items listed in "Diagnostics" on page 1063 and contact IBM Support.

00E50710

An internal error has occurred.

System action

The queue manager is terminated with this reason code. An SVC dump of the original error was requested and a record entered into SYS1.LOGREC.

System programmer response

Restart the queue manager.

Scan the SYS1.LOGREC entries looking for one or more IBM MQ errors immediately prior to the queue manager termination. If you are unable to resolve the problem, collect the items listed in “Diagnostics” on page 1063 and contact IBM Support.

00E50711

An internal error has occurred.

System action

The queue manager is terminated with this reason code. An SVC dump of the original error was requested and a record entered into SYS1.LOGREC.

System programmer response

Restart the queue manager.

Scan the SYS1.LOGREC entries looking for one or more IBM MQ errors immediately prior to the queue manager termination. If you are unable to resolve the problem, collect the items listed in “Diagnostics” on page 1063 and contact IBM Support.

00E50712

An error occurred in a latch manager function attempting to terminate the holder of an IBM MQ latch. The holder's task has been set nondispatchable by z/OS and a CALLRTM to terminate this task was unsuccessful.

System action

The queue manager is terminated with this reason code. An SVC dump of the error is requested and a record entered into SYS1.LOGREC. Register 3 at time of error contains the latch-holder's TCB address in the home address space and register 4 contains the return code from CALLRTM.

System programmer response

Restart the queue manager.

You might find the items listed in “Diagnostics” on page 1063 useful in resolving the problem. Scan the SYS1.LOGREC entries for one or more MQ errors immediately prior to the queue manager termination.

00E50713

An internal error has occurred.

System action

The queue manager is ended abnormally. An SVC dump is requested by the queue manager termination processor and a record is written to SYS1.LOGREC.

System programmer response

Restart the queue manager.

Scan the SYS1.LOGREC entries for one or more IBM MQ errors occurring immediately prior to the queue manager termination. It might be necessary to analyze the SVC dump requested. If you are unable to resolve the problem, collect the items listed in “Diagnostics” on page 1063 and contact IBM Support.

00E50715

Queue manager termination was requested following an unrecoverable error in an SRB mode execution unit.

System action

The SRB-related task was ended abnormally as a result of SRB to TCB percolation. The queue manager is ended abnormally.

System programmer response

Restart the queue manager.

You might find the items listed in "Diagnostics" on page 1063 useful in resolving the problem. Scan the SYS1.LOGREC entries for one or more IBM MQ errors occurring immediately prior to the queue manager termination.

00E50717

An internal error has occurred.

System action

The queue manager is ended abnormally.

System programmer response

Restart the queue manager.

Scan the SYS1.LOGREC entries for one or more IBM MQ errors occurring immediately prior to the queue manager termination. If an error preceded the queue manager termination request, diagnostic information can be obtained through SYS1.LOGREC and SVC dump materials. If you are unable to resolve the problem, collect the items listed in "Diagnostics" on page 1063 and contact IBM Support.

00E50719

An internal error has occurred.

System action

The queue manager is ended abnormally.

System programmer response

Restart the queue manager.

Scan the SYS1.LOGREC entries for one or more IBM MQ errors occurring immediately prior to the queue manager termination. If you are unable to resolve the problem, collect the items listed in "Diagnostics" on page 1063 and contact IBM Support.

00E50725

Queue manager termination was requested because of an unrecovered error in a scheduled SRB-mode execution unit.

System action

The SRB-related task was ended abnormally, due to SRB to TCB percolation. The queue manager is ended abnormally.

System programmer response

Restart the queue manager.

You might find the items listed in "Diagnostics" on page 1063 useful in resolving the problem. Scan the SYS1.LOGREC entries for one or more IBM MQ errors occurring immediately prior to the queue manager termination. If necessary, analyze the SVC dump requested by queue manager termination.

00E50727

A secondary error occurred during agent services functional recovery processing. This is a queue manager termination reason code.

System action

The queue manager is ended abnormally.

System programmer response

Restart the queue manager.

You might find the items listed in “Diagnostics” on page 1063 useful in resolving the problem. Scan the SYS1.LOGREC entries for one or more IBM MQ errors occurring immediately prior to the queue manager termination.

Related tasks

Collecting troubleshooting information

z/OS *Instrumentation facilities codes (X'E6')*

If an instrumentation facilities reason code occurs that is not listed here, an internal error has occurred. Collect the following diagnostic items and contact IBM Support.

Diagnostics

- The console output for the period leading up to the error.
- The system dump resulting from the error.
- A printout of SYS1.LOGREC.
- Appropriate IBM MQ, z/OS, Db2, CICS, and IMS service levels.

00E60008

An internal error has occurred.

System action

The function being traced is ended abnormally. The queue manager remains operational.

System programmer response

Collect the items listed in “Diagnostics” on page 1077 and contact IBM Support.

00E60017

This code is an internal code used by the dump formatter.

System action

The request is ended abnormally.

System programmer response

Collect the items listed in “Diagnostics” on page 1077 and contact IBM Support.

00E60085, 00E60086, 00E60087, 00E60088, 00E60089

An internal error has occurred.

System action

The request is end abnormally.

System programmer response

Collect the items listed in “Diagnostics” on page 1077 and contact IBM Support.

00E60100 through 00E60199

The reason codes X'00E60100' through X'00E60199' are used by the instrumentation facility component (IFC) when a trace event occurs for which IBM service personnel have requested a dump using the IFC selective dump service aid.

System action

The agent might be retried or terminated, depending upon the serviceability dump request.

System programmer response

The reason code is issued on the occurrence of a specified trace event. An SVC dump is taken to the SYS1.DUMPxx data set. Problem determination methods depend on the condition that IBM service personnel are attempting to trap.

00E60701

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1077 and contact IBM Support.

00E60702, 00E60703

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1077 and contact IBM Support.

Related tasks

Collecting troubleshooting information

z/OS *Distributed queuing codes (X'E7')*

If a distributed queuing reason code occurs that is not listed here, an internal error has occurred. Collect the following diagnostic items and contact IBM Support.

Diagnostics

- A description of the actions that led to the error or, if applicable, either a listing of the application program or the input string to a utility program that was being run at the time of the error.
- The channel definitions being used
- If the error affected a message channel agent, a listing of any user channel exit programs used by the message channel agent.
- The console output for the period leading up to the error.
- The queue manager job log.
- The channel initiator job log.
- The system dump resulting from the error.
- Appropriate IBM MQ, z/OS, Db2, CICS, and IMS service levels.

00E70001

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1078 and contact IBM Support.

00E70002

No adapter subtasks are active. They have failed many times and so have not been restarted.

System action

The channel initiator terminates.

System programmer response

Investigate the adapter subtask failure problems, as reported in the messages associated with each failure.

00E70003

No dispatchers are active. Either all the dispatchers failed to start, or all the dispatchers have failed many times and so have not been restarted.

System action

The channel initiator terminates.

System programmer response

Investigate the dispatcher failure problems, as reported in the messages associated with each failure.

00E70004

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1078 and contact IBM Support.

00E70007

An attempt by an adapter subtask to obtain some storage failed.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Increase the size of the channel initiator address space, or reduce the number of dispatchers, adapter subtasks, SSL server subtasks, and active channels being used.

00E70008, 00E70009, 00E7000A

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1078 and contact IBM Support.

00E70011

The channel initiator was unable to load the module CSQXBENT.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Check the console for messages indicating why CSQXBENT was not loaded. Ensure that the module is in the required library, and that it is referenced correctly.

The channel initiator attempts to load this module from the library data sets under the STEPLIB DD statement of its started task JCL procedure xxxxCHIN.

00E70013

Some adapter subtasks were requested, but none could be attached.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Investigate the adapter subtask attach problems, as reported in the messages associated with each failure. If you cannot resolve the problems, collect the items listed in “Diagnostics” on page 1078 and contact IBM Support.

00E70015

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1078 and contact IBM Support.

00E7001D

During startup, the channel initiator was unable obtain some storage below 16M.

System action

The channel initiator ends.

System programmer response

Investigate the cause of the problem.

00E7001E, 00E7001F

An internal error has occurred.

System action

The channel initiator terminates with completion code X'5C6'.

System programmer response

Restart the channel initiator.

Collect the items listed in “Diagnostics” on page 1078 and contact IBM Support.

00E70020

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Check the console for preceding error messages. If the problem cannot be resolved, collect the items listed in “Diagnostics” on page 1078 and contact IBM Support.

00E70021, 00E70022, 00E70023, 00E70024, 00E70025

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1078 and contact IBM Support.

00E70031

An internal error has occurred. A lock is currently held by a task that has terminated.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Determine why the terminated task did not free the lock. This might be due to a previous error. If you are unable to resolve the problem, collect the items listed in “Diagnostics” on page 1078 and contact IBM Support.

00E70032

An internal error has occurred. An attempt to update information held in the coupling facility failed.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1078, together with details of the queue sharing group and of the queue managers active, as well as the queue managers defined to the queue sharing group at the time. This information can be obtained by entering the following z/OS commands:

```
D XCF,GRP
```

to display a list of all queue sharing groups in the coupling facility

```
D XCF,GRP,qsg-name,ALL
```

to display status about the queue managers defined to the queue sharing group.

contact IBM Support.

00E70033

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1078 and contact IBM Support.

00E70052

No SSL server subtasks are active. They have failed many times and so have not been restarted.

System action

The channel initiator terminates.

System programmer response

Investigate the SSL server subtask failure problems, as reported in the messages associated with each failure.

00E70053

Some SSL server subtasks were requested, but none could be attached.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Investigate the SSL server subtask attach problems, as reported in the messages associated with each failure. If you cannot resolve the problems, collect the items listed in "Diagnostics" on page 1078 and contact IBM Support.

00E7010C

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in "Diagnostics" on page 1078 and contact IBM Support.

00E7010E

The dispatcher detected an inconsistency in the linkage stack.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

The most likely cause is incorrect use of the linkage stack by a user exit; exits must issue any IBM MQ API calls and return to the caller at the same linkage stack level as they were entered. If exits are not being used, or if they do not use the linkage stack, collect the items listed in "Diagnostics" on page 1078 and contact IBM Support.

00E7010F, 00E7014A

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in "Diagnostics" on page 1078 and contact IBM Support.

00E7014C

An internal error has occurred. This can be caused by the channel initiator failing to stop when running against a previous instance of the queue manager and attempting to connect to a later instance of the queue manager.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1078, terminate then restart the channel initiator and contact IBM Support.

00E7014D

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1078 and contact IBM Support.

00E7014F

An internal error has occurred. This is normally as a result of some previous error.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Check the console for preceding error messages reporting a previous error, and take the appropriate action for resolving that error. If there is no previous error, collect the items listed in “Diagnostics” on page 1078 and contact IBM Support.

00E7015A, 00E70214, 00E70216, 00E70226, 00E70231, 00E70232, 00E70233, 00E70501, 00E70522, 00E70543, 00E70546, 00E70553

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1078 and contact IBM Support.

00E70054, 00E70055, 00E70056

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1078 and contact IBM Support.

00E70057, 00E70058

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1078 and contact IBM Support.

00E70708

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in “Diagnostics” on page 1078 and contact IBM Support.

00E70802

No cluster workload exit server subtasks are active. They have failed many times, and so have not been restarted.

System action

Cluster workload exit services are disabled.

System programmer response

Investigate the cluster workload exit server subtask failure problems, as reported in the messages associated with each failure.

00E7080B

A cluster workload user exit did not return to the queue manager within the allowed time, as specified by the **EXITLIM** system parameter. The task running the exit is terminated with this reason. This code is preceded by message CSQV445E.

System action

Processing continues. The cluster destination is chosen using the usual algorithm.

System programmer response

Investigate why the user exit did not complete in time. Message CSQV445E displays the name of the exit program.

Related tasks

Collecting troubleshooting information

z/OS *Initialization procedure and general services codes (X'E8')*

If an initialization procedure reason code occurs that is not listed here, an internal error has occurred. Collect the following diagnostic items and contact IBM Support.

Diagnostics

- The console output for the period leading up to the error.
- The queue manager job log.
- The system dump resulting from the error.
- A printout of SYS1.LOGREC.
- The system parameter load module.
- The initialization procedure.
- The started task JCL procedure for this queue manager.
- Appropriate IBM MQ, z/OS, Db2, CICS, and IMS service levels.

00E80001

An internal error has occurred.

System action

The queue manager is terminated. A record is written to SYS1.LOGREC, and an SVC dump is requested.

System programmer response

Restart the queue manager.

Collect the items listed in “Diagnostics” on page 1084 and contact IBM Support.

00E80002

The queue manager address space was not started correctly or an error occurred during z/OS IEFSSREQ processing.

System action

The queue manager is terminated. A record is written to SYS1.LOGREC, and an SVC dump is requested. Register 9 contains the address of an 8-byte field that contains the following diagnostic information:

- Bytes 1 through 4 - subsystem name
- Bytes 5 through 8 - contents of register 15 that contains the return code set by the z/OS IEFSSREQ macro

System programmer response

You might find the items listed in "Diagnostics" on page 1084 useful in resolving the problem.

00E80003, 00E80004, 00E80005, 00E80006

An internal error has occurred.

System action

A record is written to SYS1.LOGREC, and an SVC dump is requested.

System programmer response

Collect the items listed in "Diagnostics" on page 1084 and contact IBM Support.

00E8000E

An ESTAE could not be established for the queue manager address space control task.

System action

The queue manager is terminated. A record is written to SYS1.LOGREC, and an SVC dump is requested. Register 9 contains the address of a 4-byte field that contains the ESTAE macro return code.

System programmer response

Restart the queue manager.

You might find the items listed in "Diagnostics" on page 1084 useful in resolving the problem.

00E8000F

Invalid startup parameters were specified. This was probably caused by an attempt to start the queue manager by some means other than a **START QMGR** command.

System action

The queue manager is terminated. A record is written to SYS1.LOGREC, and an SVC dump is requested.

System programmer response

Restart the queue manager.

If you are unable to resolve the problem, collect the items listed in "Diagnostics" on page 1084 and contact IBM Support.

00E80010

An invalid product was specified. This abend is preceded by one or more instances of message CSQY038E. See this message for more details.

System action

The queue manager is terminated.

System programmer response

Locate the related CSQY038E messages for the queue manager, and correct the issue described in each of those messages.

00E80011

The address space could not be made non-swappable.

System action

The queue manager is terminated. A record is written to SYS1.LOGREC, and an SVC dump is requested.

System programmer response

Restart the queue manager.

You might find the items listed in “Diagnostics” on page 1084 useful in resolving the problem.

00E80012

An internal error has occurred.

System programmer response

Collect the items listed in “Diagnostics” on page 1084 and contact IBM Support.

00E80013, 00E8001F, 00E8002F

An internal error has occurred.

System action

The queue manager is terminated. A record is written to SYS1.LOGREC, and an SVC dump is requested.

System programmer response

Restart the queue manager.

Collect the items listed in “Diagnostics” on page 1084 and contact IBM Support.

00E80031

An unsupported input parameter was detected for allied address space initialization.

System action

The caller's task is ended abnormally. A record is written to SYS1.LOGREC, and an SVC dump is requested.

System programmer response

Collect the items listed in “Diagnostics” on page 1084 and contact IBM Support.

00E80032

An unsupported input parameter was detected for allied address space termination.

System action

The caller's task is ended abnormally. A record is written to SYS1.LOGREC, and an SVC dump is requested.

System programmer response

Collect the items listed in “Diagnostics” on page 1084 and contact IBM Support.

00E80033

This reason code accompanies a X'6C6' completion code. This module detected that the queue manager was terminating.

System action

The caller's task is ended abnormally with code X'6C6'. A record is written to SYS1.LOGREC, and an SVC dump is requested.

System programmer response

You might find the items listed in “Diagnostics” on page 1084 useful in resolving the problem.

00E8003C

An internal error has occurred.

System action

The caller's task is ended abnormally. A record is written to SYS1.LOGREC, and an SVC dump is requested.

System programmer response

Collect the items listed in “Diagnostics” on page 1084 and contact IBM Support.

00E8003D

An internal error has occurred.

System action

Abnormal termination of the queue manager is initiated. A record is written to SYS1.LOGREC, and an SVC dump is requested.

System programmer response

Restart the queue manager.

Collect the items listed in “Diagnostics” on page 1084 and contact IBM Support.

00E8003E

An ESTAE could not be established in an address space about to be initialized as an IBM MQ allied address space.

System action

The caller's task is ended abnormally. A record is written to SYS1.LOGREC, and an SVC dump is requested.

System programmer response

If you are unable to resolve the problem, collect the items listed in “Diagnostics” on page 1084 and contact IBM Support.

00E8003F

An internal error has occurred.

System action

The caller's task is ended abnormally. A record is written to SYS1.LOGREC, and an SVC dump is requested.

System programmer response

Collect the items listed in “Diagnostics” on page 1084 and contact IBM Support.

00E80041

An internal error has occurred.

System action

The queue manager is terminated. A record is written to SYS1.LOGREC, and an SVC dump is requested.

System programmer response

Restart the queue manager.

Collect the items listed in “Diagnostics” on page 1084 and contact IBM Support.

00E80042, 00E8004F

An internal error has occurred.

System action

The queue manager is terminated. A record is written to SYS1.LOGREC, and an SVC dump is requested.

System programmer response

Restart the queue manager.

Collect the items listed in “Diagnostics” on page 1084 and contact IBM Support.

00E80051

An error was detected in the command that was used to start the queue manager.

System action

The queue manager is terminated.

System programmer response

Reenter the command if it was entered incorrectly.

If you are unable to resolve the problem, contact IBM Support.

00E80052, 00E80053, 00E80054, 00E80055

An internal error has occurred.

System action

The queue manager is terminated. A record is written to SYS1.LOGREC, and an SVC dump is requested.

System programmer response

Restart the queue manager.

Collect the items listed in “Diagnostics” on page 1084 and contact IBM Support.

00E80057

An error occurred while trying to start a queue manager address space. One possible cause of this problem is an error in the started task JCL procedure for the queue manager.

System action

The queue manager is terminated. A record is written to SYS1.LOGREC, and an SVC dump is requested.

System programmer response

Restart the queue manager.

You might find the items listed in “Diagnostics” on page 1084 useful in resolving the problem.

00E80058

An error occurred during command prefix registration.

System action

The queue manager ends abnormally.

System programmer response

See the accompanying CSQYxxx messages for information about the cause of the problem.

Restart the queue manager after correcting the problem.

00E8005F, 00E80061, 00E8006F, 00E8007F

An internal error has occurred.

System action

The queue manager is terminated. A record is written to SYS1.LOGREC, and an SVC dump is requested.

System programmer response

Restart the queue manager.

Collect the items listed in "Diagnostics" on page 1084 and contact IBM Support.

00E80081

An invalid load module was detected.

System action

The queue manager is terminated. A record is written to SYS1.LOGREC, and an SVC dump is requested. Register 9 contains the address of an 8-byte field that contains the name of the module in error.

System programmer response

Check that the installation process was successful.

Restart the queue manager after resolving the problem.

If you are unable to resolve the problem, collect the items listed in "Diagnostics" on page 1084 and contact IBM Support.

00E80084

A resource manager provided notification of an error during queue manager startup notification processing.

System action

The queue manager is terminated. A record is written to SYS1.LOGREC, and an SVC dump is requested. Register 9 contains the address of a 4-byte field that contains the RMID of the resource manager that requested queue manager termination.

System programmer response

Look for error messages indicating the cause of the problem.

Restart the queue manager after resolving the problem.

If you are unable to solve the problem, collect the items listed in "Diagnostics" on page 1084, together with the contents of the BSDS and a GTF trace, and contact IBM Support.

00E8008F, 00E80091, 00E8009F, 00E800AF, 00E800B1

An internal error has occurred.

System action

The queue manager is terminated. A record is written to SYS1.LOGREC, and an SVC dump is requested.

System programmer response

Restart the queue manager.

Collect the items listed in "Diagnostics" on page 1084 and contact IBM Support.

00E800B2

The queue manager initialization procedure found that the version of ZPARM loaded was compiled for a higher release of IBM MQ.

System action

Startup is terminated.

System programmer response

Check whether the correct ZPARM has been loaded at initialization.

The existence of this version of ZPARM implies that the queue manager has been running at a higher release of the product.

Check to see if the queue manager has been started on a higher release of the product. If this is the case, you have inadvertently started IBM MQ with the wrong version of the product libraries.

It might still be possible to rebuild ZPARM using the macros from SCSQMACS for the current release of the product.

00E800CE

An ESTAE could not be established.

System action

The queue manager is terminated. A record is written to SYS1.LOGREC, and an SVC dump is requested. Register 9 contains the address of a 4-byte field that contains the ESTAE macro return code.

System programmer response

Restart the queue manager.

You might find the items listed in “Diagnostics” on page 1084 useful in resolving the problem.

00E800D1

An internal error has occurred.

System action

The queue manager is terminated. A record is written to SYS1.LOGREC, and an SVC dump is requested.

System programmer response

Restart the queue manager.

Collect the items listed in “Diagnostics” on page 1084 and contact IBM Support.

00E800D2

An error was encountered while attempting to obtain the z/OS LOCAL lock.

System action

The queue manager is terminated. A record is written to SYS1.LOGREC, and an SVC dump is requested.

System programmer response

Restart the queue manager.

You might find the items listed in “Diagnostics” on page 1084 useful in resolving the problem.

00E800D3

An error was encountered while attempting to release the z/OS LOCAL lock.

System action

The queue manager is terminated. A record is written to SYS1.LOGREC, and an SVC dump is requested.

System programmer response

Restart the queue manager.

You might find the items listed in “Diagnostics” on page 1084 useful in resolving the problem.

00E800DF

An internal error has occurred.

System action

The queue manager is terminated. A record is written to SYS1.LOGREC, and an SVC dump is requested.

System programmer response

Restart the queue manager.

Collect the items listed in “Diagnostics” on page 1084 and contact IBM Support.

00E80100

The queue manager was ended abnormally because the queue manager address space control task ESTAE was entered. This reason code is issued for all completion codes, except for the X'5C6' completion code.

The queue manager is unable to determine the cause of the error.

System action

Termination of the queue manager is initiated. A record is written to SYS1.LOGREC, and an SVC dump is requested.

System programmer response

Restart the queue manager after resolving the problem.

The subcomponent that caused the error is unknown. This reason code might be returned if the queue manager is unable to find the system parameter load module you specified on the **START QMGR** command (the default name is CSQZPARM). Check that the module you specified is available.

This reason code is also issued if the queue manager is canceled by the z/OS command **CANCEL**. If this is the case, determine why the queue manager was canceled.

You might find the items listed in “Diagnostics” on page 1084, together with the contents of the BSDS and a GTF trace, useful in resolving the problem.

00E8011D

An internal error has occurred.

System action

Termination of queue manager is initiated. A record is written to SYS1.LOGREC, and an SVC dump is requested.

System programmer response

Restart the queue manager.

Collect the items listed in “Diagnostics” on page 1084 and contact IBM Support.

00E8011E

The allied address space task primary ESTAE detected that the secondary ESTAE could not be established.

System action

Abnormal termination of allied address space is continued. A record is written to SYS1.LOGREC, and an SVC dump is requested.

System programmer response

You might find the items listed in “Diagnostics” on page 1084 useful in resolving the problem.

00E8011F

The allied address space task primary ESTAE was entered without a subsystem diagnostic work area (SDWA) provided by z/OS RTM.

System action

Abnormal termination of the allied address space is continued. A record is written to SYS1.LOGREC, and an SVC dump is requested.

System programmer response

You might find the items listed in “Diagnostics” on page 1084 useful in resolving the problem.

00E8012D

An internal error has occurred.

System action

Abnormal termination of queue manager is initiated. A record is written to SYS1.LOGREC, and an SVC dump is requested.

System programmer response

Restart the queue manager.

Collect the items listed in “Diagnostics” on page 1084 and contact IBM Support.

00E8012F

The allied address space task secondary ESTAE was entered without a subsystem diagnostic work area (SDWA) provided by z/OS .

System action

Continue with the abnormal termination of the allied address space. A record is written to SYS1.LOGREC, and an SVC dump is requested.

System programmer response

You might find the items listed in “Diagnostics” on page 1084 useful in resolving the problem.

00E80130

The FRR that protects the **START QMGR**/**STOP QMGR** command processor function was entered while a valid **STOP QMGR** command was being processed.

System action

The queue manager is terminated. A record is written to SYS1.LOGREC, and an SVC dump is requested.

System programmer response

You might find the items listed in “Diagnostics” on page 1084 useful in resolving the problem.

00E80140

An internal error has occurred.

System action

The queue manager is terminated. A record is written to SYS1.LOGREC, and an SVC dump is requested.

System programmer response

Restart the queue manager.

Collect the items listed in “Diagnostics” on page 1084 and contact IBM Support.

00E80150, 00E80151

An invalid module was detected.

System action

The queue manager is terminated. A record is written to SYS1.LOGREC, and an SVC dump is requested. Register 9 contains the address of a 12-byte field that contains the following diagnostic information:

- Bytes 1 through 8 contain the name of the load module that contains the initialization entry point list with the invalid entry

System programmer response

Restart the queue manager after resolving the problem.

Check that the installation process was successful. If you are unable to resolve the problem, collect the items listed in “Diagnostics” on page 1084 and contact IBM Support.

00E8015F

An internal error has occurred.

System action

The queue manager is terminated. A record is written to SYS1.LOGREC, and an SVC dump is requested.

System programmer response

Restart the queue manager.

Collect the items listed in “Diagnostics” on page 1084 and contact IBM Support.

00E80160

The queue manager initialization procedures found that a load module had an invalid AMODE or RMODE attribute.

System action

Queue manager startup is terminated.

System programmer response

See message CSQY006E.

00E80161

The queue manager initialization procedures found that a load module was not at the correct level for the version of the queue manager that was being started.

System action

Queue manager startup is terminated.

System programmer response

See message CSQY010E.

00E80162

The queue manager initialization procedures found that the storage protect key was not 7. The most likely cause is that the program properties table (PPT) entry for CSQYASCP has not been specified correctly.

System action

Queue manager startup is terminated.

System programmer response

Restart the queue manager after resolving the problem.

For information about specifying the PPT entry for CSQYASCP, see Update the z/OS program properties table.

00E80163

The queue manager initialization procedures found that they were not APF authorized. The most likely cause is that one or more of the data sets in the //STEPLIB concatenation is not APF authorized.

System action

Queue manager startup is terminated.

System programmer response

Restart the queue manager after resolving the problem.

For information about APF authorization for the IBM MQ load libraries, see APF authorize the IBM MQ load libraries

00E80170

An internal error has occurred.

System action

The request is ignored.

System programmer response

Collect the items listed in “Diagnostics” on page 1084 and contact IBM Support.

00E80171

The queue manager terminated because you replied to the message CSQY041D WTOR with the letter N.

System programmer response

See message CSQY041D for more information.

00E80172

The queue manager terminated because you replied to the message CSQY044D WTOR with the letter N.

System programmer response

See message CSQY044D for more information.

V 10.0.0 **00E80192**

The queue manager initialization procedures found that a required control block had an incorrect length.

System action

Queue manager startup is terminated.

System programmer response

Ensure that each early code load module in the Link Pack Area (LPA) is at the same level. If you are unable to resolve the problem, collect the items listed in “Diagnostics” on page 1084 and contact IBM Support.

Related tasks

Collecting troubleshooting information

z/OS *System parameter manager codes (X'E9')*

If a system parameter manager reason code occurs that is not listed here, an internal error has occurred. Collect the following diagnostic items and contact IBM Support.

Diagnostics

- The console output for the period leading up to the error.
- The queue manager job log.
- The system dump resulting from the error.
- A printout of SYS1.LOGREC.
- The system parameter load module.
- The initialization procedure.
- The started task JCL procedure for this queue manager.
- Appropriate IBM MQ, z/OS, Db2, CICS, and IMS service levels.

00E90101

An error has occurred while trying to open IBM MQ resources. The most likely cause is that a customized system parameter load module specified on the **START QMGR** command is not available.

System action

A record is written to SYS1.LOGREC, and an SVC dump is requested.

System programmer response

Check that the system parameter load module you specified on the **START QMGR** command (the default name is CSQZPARM) is available for use. If it is, collect the items listed in "Diagnostics" on page 1095 and contact IBM Support.

00E90201

An internal error has occurred while attempting to open IBM MQ resources.

System action

A record is written to SYS1.LOGREC, and an SVC dump is requested.

System programmer response

Collect the items listed in "Diagnostics" on page 1095 and contact IBM Support.

00E90202

An error has occurred while attempting to open IBM MQ resources. The most likely cause is that a customized system parameter load module specified on the START QMGR command (the default name is CSQZPARM) has been built incorrectly.

System action

A record is written to SYS1.LOGREC, and an SVC dump is requested.

System programmer response

Check that the system parameter load module that you specified is available, and that it was linked correctly. See CSQ4ZPRM for sample link-edit JCL. and for information about the system parameter modules, see Tailor your system parameter module.

Restart the queue manager. If the problem persists, collect the items listed in "Diagnostics" on page 1095 and contact IBM Support.

00E90203

An internal error has occurred while attempting to verify descriptor control information in IBM MQ resources.

System action

A record is written to SYS1.LOGREC, and an SVC dump is requested.

System programmer response

Collect the items listed in “Diagnostics” on page 1095 and contact IBM Support.

00E90301

An internal error has occurred while attempting to close IBM MQ resources.

System action

A record is written to SYS1.LOGREC, and an SVC dump is requested.

System programmer response

Collect the items listed in “Diagnostics” on page 1095 and contact IBM Support.

Related tasks

Collecting troubleshooting information

z/OS *Service facilities codes (X'F1')*

00F10001, 00F10002, 00F10003, 00F10004, 00F10005, 00F10006, 00F10007, 00F10008, 00F10009, 00F10010, 00F10011, 00F10012, 00F10013, 00F10014, 00F10015, 00F10016, 00F10017, 00F10018, 00F10019, 00F1001B, 00F1001C

An internal error has been detected in the CSQ1LOGP log print utility.

System action

A dump is requested. The utility ends abnormally with completion code X'5C6'.

System programmer response

Collect the following diagnostic items and contact IBM Support:

- Utility report output
- System dump resulting from the error, if any
- Appropriate IBM MQ, z/OS, Db2, CICS, and IMS service levels.

00F1001A

The IBM MQ utility program was unable to initialize the required data conversion services. The reason for the failure is in register 0:

00C10002, 00C1000E
Unable to load modules

00C10003
Insufficient storage

Other
Internal error

System action

A dump is requested. The utility ends abnormally with completion code X'5C6'.

System programmer response

Check the console for messages indicating that a module was not loaded. Ensure that the module is in the required library (SCSQAUTH), that it is referenced correctly, and resubmit the job.

The utility program attempts to load this module from the library data sets under the STEPLIB DD statement of the utility address space.

If you are unable to resolve the problem, contact IBM Support.

00F1001D

The IBM MQ utility program was unable to allocate storage required to perform the request.

System action

A dump is requested. The utility ends abnormally with completion code X'5C6'.

System programmer response

It is probable that the REGION parameter on the EXEC statement of the job control language (JCL) for this invocation is too small. Increase the REGION size, and resubmit the request.

If you are unable to resolve the problem, contact IBM Support.

00F10100

An internal error has been detected in the CSQ1LOGP log print utility.

System action

A dump is requested. The utility ends abnormally with completion code X'5C6'.

System programmer response

Resubmit the job.

contact IBM Support if the problem persists.

00F10101

The stand-alone log read function returned an invalid RBA. See the explanation for message CSQ1211E.

System action

A dump is requested. The utility ends abnormally with completion code X'5C6'.

System programmer response

If you determine that the data set is a log data set and that it is not damaged, contact IBM Support.

Related tasks

Collecting troubleshooting information

z/OS *IBM MQ-IMS bridge codes (X'F2')*

If an IBM MQ-IMS bridge reason code occurs that is not listed here, an internal error has occurred. Collect the following diagnostic items and contact IBM Support.

Diagnostics

- A description of the actions that led to the error or, if applicable, either a listing of the application program or the input string to a utility program that was being run at the time of the error.
- The console output for the period leading up to the error.
- The queue manager job log.
- The IMS job logs.
- The system dump resulting from the error.
- Appropriate IBM MQ, z/OS, Db2, CICS, and IMS service levels.

00F20001, 00F20002, 00F20003, 00F20004, 00F20005, 00F20006, 00F20007, 00F20008, 00F20009, 00F2000A, 00F2000B, 00F2000C, 00F2000D, 00F2000E, 00F2000F, 00F20010, 00F20011

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in "Diagnostics" on page 1097 and contact IBM Support.

00F20012

The IBM MQ-IMS bridge received a bad return code from IXCQUERY macro.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Registers 3 and 4 contain the return and reason codes from XCF. Refer to the *MVS Programming: Sysplex Services Reference* for information about these codes.

00F20013

The IBM MQ-IMS bridge received a bad return from IXCJOIN macro.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Registers 3 and 4 contain the return and reason codes from XCF. Refer to the *MVS Programming: Sysplex Services Reference* for information about these codes.

00F20014

The IBM MQ-IMS bridge received a bad return from IXCCREAT macro.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Registers 3 and 4 contain the return and reason codes from XCF. Refer to the *MVS Programming: Sysplex Services Reference* for information about these codes.

Use the IMS DIS OTMA command to see if the OTMACON member name is already in use. This can be caused by specifying the IMS system instead of the queue manager name in the OTMACON member name.

00F20015, 00F20016

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in "Diagnostics" on page 1097 and contact IBM Support.

00F20017

The IBM MQ-IMS bridge received a bad return from IXCLEAVE macro.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Registers 3 and 4 contain the return and reason codes from XCF. Refer to the *MVS Programming: Sysplex Services Reference* for information about these codes.

00F20018

The IBM MQ-IMS bridge received a bad return from IXCDELET macro.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Registers 3 and 4 contain the return and reason codes from XCF. Refer to the *MVS Programming: Sysplex Services Reference* for information about these codes. contact IBM Support to report the problem.

00F20019, 00F2001A, 00F2001B, 00F2001C, 00F2001D, 00F2001E, 00F2001F, 00F20020, 00F20021, 00F20022

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in "Diagnostics" on page 1097 and contact IBM Support.

00F20023

The IBM MQ-IMS bridge received a bad return code from IXCMSGO.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Registers 2 and 3 contain the return and reason codes from XCF. Refer to the *MVS Programming: Sysplex Services Reference* for information about these codes.

00F20024, 00F20026, 00F20027, 00F20029, 00F2002A, 00F2002B

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in "Diagnostics" on page 1097 and contact IBM Support.

00F2002C

The IBM MQ-IMS bridge received a bad return code from IXCMSGO.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Registers 2 and 3 contain the return and reason codes from XCF. Refer to the *MVS Programming: Sysplex Services Reference* for information about these codes.

00F2002D, 00F2002E

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 1097 and contact IBM Support.

00F20030

The IBM MQ-IMS bridge received a bad return code from IXCMSGO.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Registers 2 and 3 contain the return and reason codes from XCF. Refer to the *MVS Programming: Sysplex Services Reference* for information about these codes.

00F20031

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 1097 and contact IBM Support.

00F20032

The IBM MQ-IMS bridge received a bad return code from IXCMSGO.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Registers 2 and 3 contain the return and reason codes from XCF. Refer to the *MVS Programming: Sysplex Services Reference* for information about these codes.

00F20035, 00F20036, 00F20037, 00F20038, 00F20039, 00F2003A, 00F2003B, 00F2003D, 00F2003E, 00F2003F, 00F20040

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 1097 and contact IBM Support.

00F20041

The IBM MQ-IMS bridge received an MQOPEN error.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

contact IBM Support to report the problem.

00F20042

The IBM MQ-IMS bridge received an MQCLOSE error.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

contact IBM Support to report the problem.

00F20043

The IBM MQ-IMS bridge received an MQGET error.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

contact IBM Support to report the problem.

00F20044

The IBM MQ-IMS bridge received an MQPUT error.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

contact IBM Support to report the problem.

00F20045

The IBM MQ-IMS bridge received an MQOPEN error.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

contact IBM Support to report the problem.

00F20046

The IBM MQ-IMS bridge received an MQCLOSE error.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

contact IBM Support to report the problem.

00F20047

The IBM MQ-IMS bridge received an MQGET error.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

contact IBM Support to report the problem.

00F20048

The IBM MQ-IMS bridge received an MQPUT error.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

contact IBM Support to report the problem.

00F20049

The IBM MQ-IMS bridge received an MQPUT1 error.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

contact IBM Support to report the problem.

00F2004A, 00F2004B, 00F2004C, 00F2004D, 00F2004E, 00F2004F, 00F20050, 00F20051, 00F20052, 00F20053, 00F20054, 00F20055, 00F20056, 00F20057

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

Collect the items listed in “Diagnostics” on page 1097 and contact IBM Support.

00F20058

The IBM MQ-IMS bridge received an MQPUT1 error.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

contact IBM Support to report the problem.

00F20059

The IBM MQ-IMS bridge received a severe sense code in an IMS negative response.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

The IMS sense code is given in message CSQ2003I..

00F20069

The IBM MQ-IMS bridge received an error when trying to resolve an in-doubt unit of recovery.

System action

The current execution unit terminates with completion code X'5C6', and a dump is produced.

System programmer response

contact IBM Support to report the problem.

Related tasks

Collecting troubleshooting information

z/OS *Subsystem support codes (X'F3')*

Many of the following reason codes are returned in register 15 at the time of an abnormal termination with completion code X'0Cx', and not as the reason code for a completion code of X'5C6'. This is indicated in the descriptions that follow.

If a subsystem support reason code occurs that is not listed here, an internal error has occurred. Collect the following diagnostic items and contact IBM Support.

Diagnostics

- A description of the actions that led to the error or, if applicable, either a listing of the application program or the input string to a utility program that was being run at the time of the error.
- The console output for the period leading up to the error.
- The queue manager job log.
- The system dump resulting from the error.
- A printout of SYS1.LOGREC.
- Appropriate IBM MQ, z/OS, Db2, CICS, and IMS service levels.

00F30003, 00F30004, 00F30005

An internal error has occurred.

System action

The request is not processed. A dump is taken, and an entry is written in SYS1.LOGREC.

System programmer response

Collect the items listed in “Diagnostics” on page 1103 and contact IBM Support.

00F30006

An internal error has occurred.

System action

The request is not processed.

System programmer response

Collect the items listed in “Diagnostics” on page 1103 and contact IBM Support.

00F30007, 00F30008

An internal error has occurred.

System action

The request is not processed. A dump is taken, and an entry is written in SYS1.LOGREC.

System programmer response

Collect the items listed in “Diagnostics” on page 1103 and contact IBM Support.

00F30014

An internal error has occurred.

System action

The requester's task is ended abnormally with completion code X'5C6'. A dump is taken, and an entry is written in SYS1.LOGREC.

System programmer response

Collect the items listed in “Diagnostics” on page 1103 and contact IBM Support.

00F30027, 00F30030 ,00F30032, 00F30033, 00F30038

An internal error has occurred.

System action

The request is not processed. A dump is taken, and an entry is written in SYS1.LOGREC.

System programmer response

Collect the items listed in “Diagnostics” on page 1103 and contact IBM Support.

00F30042

An internal error has occurred.

System action

A dump is taken, and an entry is written in SYS1.LOGREC.

System programmer response

Collect the items listed in “Diagnostics” on page 1103 and contact IBM Support.

00F30048

An internal error has occurred.

System action

The request is not processed. A dump is taken, and an entry is written in SYS1.LOGREC.

System programmer response

Collect the items listed in “Diagnostics” on page 1103 and contact IBM Support.

00F30052

The recovery coordinator for the caller has already terminated, so the connection from the caller to IBM MQ has been terminated.

System action

The request is not processed. The connection from the caller to IBM MQ is terminated.

The caller might reconnect to IBM MQ when the recovery coordinator has been restarted.

System programmer response

Identify and restart the recovery coordinator.

This abnormal termination is most commonly associated with a termination of RRS. There might be additional CSQ3009E messages on the console log associated with the termination of RRS.

00F30053

An internal error has occurred.

System action

The request is not processed. A dump is taken, and an entry is written in SYS1.LOGREC.

System programmer response

Collect the items listed in “Diagnostics” on page 1103 and contact IBM Support.

00F30067

An internal error has occurred.

System action

The connection request is not processed. A dump is taken, and an entry is written in SYS1.LOGREC.

System programmer response

Collect the items listed in “Diagnostics” on page 1103 and contact IBM Support.

00F30070

Functional recovery for the connection processing could not be established. The executing module could not establish its ESTAE. This can occur if the current address space has insufficient storage. This might lead to an abnormal termination of the queue manager.

System action

The connection request is not processed. The caller is ended abnormally with completion code X'5C6' and this reason code.

System programmer response

Restart the queue manager if necessary. A dump should be taken for problem analysis.

Examine the usage and free areas in the LSQA portion of the current address space private area. If necessary, have the size of the private areas expanded.

The caller should produce a SYS1.LOGREC entry and an SVC dump, so that you can examine the LSQA area. You might find the items listed in “Diagnostics” on page 1103 useful in resolving the problem.

00F30071

An internal error has occurred.

System action

The connection request is not processed. A dump is taken, and an entry is written in SYS1.LOGREC.

System programmer response

Collect the items listed in “Diagnostics” on page 1103 and contact IBM Support.

00F30075

An internal error has occurred.

System action

A dump is taken, and an entry is written in SYS1.LOGREC.

System programmer response

Collect the items listed in “Diagnostics” on page 1103 and contact IBM Support.

00F30078

An internal error has occurred.

System action

The request is not processed. A dump is taken, and an entry is written in SYS1.LOGREC.

System programmer response

Collect the items listed in “Diagnostics” on page 1103 and contact IBM Support.

00F30080

An internal error has occurred.

System action

The application program is ended abnormally with completion code X'5C6' and this reason code. A dump is taken, and an entry is written in SYS1.LOGREC.

System programmer response

Collect the items listed in “Diagnostics” on page 1103 and contact IBM Support.

00F30091

The application program issued an RRSAF IDENTIFY function request, but RRS is not available.

System action

The IDENTIFY request is not processed.

00F30093

The application program issued an RRSAF TERMINATE THREAD or TERMINATE IDENTIFY function request, but the application has issued an IBM MQ API request since the last invocation of SRRCMIT or SRRBACK and therefore is not at a point of consistency.

System action

The function request is not processed.

00F30095

An internal error was detected in either MQ or RRS.

System action

The application is ended abnormally. The error is recorded in the SYS1.LOGREC data set and an SVC dump is requested.

This error might, in many cases, eventually cause the queue manager to terminate abnormally.

System programmer response

This is probably either an error in IBM MQ or in RRS.

Collect the items listed in "Diagnostics" on page 1103 and contact IBM Support.

00F30096

An internal error was detected in either IBM MQ or RRS Context Services.

System action

The application is ended abnormally. The error is recorded in the SYS1.LOGREC data set and an SVC dump is requested.

This error might, in many cases, eventually cause the queue manager to terminate abnormally.

System programmer response

This is probably either an error in IBM MQ or in RRS.

Collect the items listed in "Diagnostics" on page 1103 and contact IBM Support.

00F30101

The parameter contained in the IEFSSNxx member used to initialize IBM MQ (and other subsystems) is in error. See message CSQ3101E for details.

System action

See message CSQ3101E.

System programmer response

See message CSQ3101E.

You might find the items listed in "Diagnostics" on page 1103 useful in resolving the problem.

00F30102

The parameter contained in the IEFSSNxx member used to initialize IBM MQ (and other subsystems) is in error. The IBM MQ command prefix (CPF) must not be blank. For details, see message CSQ3102E.

System action

See message CSQ3102E.

System programmer response

See message CSQ3102E.

You might find the items listed in "Diagnostics" on page 1103 useful in resolving the problem.

00F30103

The parameter contained in the IEFSSNxx member used to initialize IBM MQ (and other subsystems) is in error or the named module is not resident in a library available during IPL. See message CSQ3103E for details.

System action

See message CSQ3103E.

System programmer response

See message CSQ3103E.

You might find the items listed in "Diagnostics" on page 1103 useful in resolving the problem.

00F30104

Module CSQ3UR00 was unable to obtain the affinity table index for the named subsystem. z/OS did not recognize the named subsystem. See message CSQ3109E for details.

System action

See message CSQ3109E.

System programmer response

See message CSQ3109E.

You might find the items listed in "Diagnostics" on page 1103 useful in resolving the problem.

00F30105

Module CSQ3UR00 was unable to load Early module CSQ3EPX. Either there was an I/O error, or the named module is not resident in a library available during IPL. See message CSQ3105E for details.

System action

See message CSQ3105E.

System programmer response

See message CSQ3105E.

You might find the items listed in "Diagnostics" on page 1103 useful in resolving the problem.

00F30106

The parameter contained in the IEFSSNxx member used to initialize IBM MQ (and other subsystems) is in error. The scope of the IBM MQ command prefix (CPF) is not valid. For details, see message CSQ3112E.

System action

See message CSQ3112E.

System programmer response

See message CSQ3112E.

You might find the items listed in "Diagnostics" on page 1103 useful in resolving the problem.

00F30107

An error occurred during command prefix registration.

System action

The IBM MQ subsystem ends abnormally.

System programmer response

See the accompanying CSQ3xxx messages for information about the cause of the problem.

00F30112

A level mismatch was detected between early code modules CSQ3INI and CSQ3EPX.

System action

The IBM MQ subsystem ends abnormally.

System programmer response

Ensure that each early code load module in the Link Pack Area (LPA) is at the same level. If you are unable to resolve the problem, collect the items listed in "Diagnostics" on page 1103 and contact IBM Support.

00F30210, 00F30211, 00F30212, 00F30213, 00F30214

An internal error has occurred.

System action

The caller is ended abnormally. An SVC dump and associated SYS1.LOGREC entries are produced.

System programmer response

Collect the items listed in "Diagnostics" on page 1103 and contact IBM Support.

00F30216

An attempt to create a queue manager address space failed. This is probably because the user who issued the **START QMGR** command has insufficient authority.

System action

The current **START** command processing is terminated. An SVC dump and associated SYS1.LOGREC entries are produced.

System programmer response

Check the authority of users and consoles to issue commands. Retry the command.

You might find the items listed in "Diagnostics" on page 1103 useful in resolving the problem.

00F30217

The console ID for the z/OS console that entered the current command is not found in the z/OS unit control module (UCM) structure. An internal z/OS command might have been incorrectly issued by an application program that provided invalid input parameters.

System action

The caller is ended abnormally.

System programmer response

Retry the START QMGR command. If the command was unsuccessful, collect the items listed in "Diagnostics" on page 1103 and contact IBM Support.

00F30218

An internal error has occurred.

System action

The current task is ended abnormally. The calling task might have requested an SVC dump or created associated SYS1.LOGREC entries.

System programmer response

Collect the items listed in "Diagnostics" on page 1103 and contact IBM Support.

00F30219

An internal error has occurred.

System action

The calling task is ended abnormally. The calling task might have requested an SVC dump or created associated SYS1.LOGREC entries.

System programmer response

Cancel the queue manager. End-of-task processing might still work, and it does a more complete clean-up than end-of-memory processing does. If this does not work, issue the z/OS command **FORCE** for the queue manager. If the problem is still unresolved, it might be necessary to perform an IPL of your z/OS system.

Collect the items listed in "Diagnostics" on page 1103 and contact IBM Support.

00F3021A

An internal error has occurred.

System action

The calling task is ended abnormally. An SVC dump and associated SYS1.LOGREC entries are produced.

System programmer response

Stop the queue manager and reissue the **START QMGR** command.

Collect the items listed in "Diagnostics" on page 1103 and contact IBM Support.

00F3021C

An ESTAE could not be established. This can occur if the z/OS system address space that is broadcasting the command has insufficient storage.

System action

The caller is ended abnormally (without a dump). The current START command processing is terminated.

System programmer response

Retry the command. If the error persists, it might be necessary to perform an IPL of your z/OS system.

Examine the LOGREC entries, and the console log for indications of a z/OS error, and try increasing the storage.

If you are unable to resolve the problem, collect the items listed in "Diagnostics" on page 1103 and contact IBM Support.

00F3021D

An ESTAE could not be established during either the initialization or termination of the queue manager.

This can occur during initialization if the z/OS system address space that is broadcasting the first command (assumed to be the **START** command) has insufficient storage.

This can occur during termination if the current address space (usually the queue manager, or in the case of EOM broadcast, a z/OS system address space) has insufficient storage.

System action

The caller is ended abnormally without taking a system dump. The initialization stops, but termination proceeds.

System programmer response

Retry the command after the queue manager has terminated. If the problem persists, it might be necessary to perform an IPL of your z/OS system.

Examine the LOGREC entries, and the console log for indications of a z/OS error, and try increasing the storage.

If you are unable to resolve the problem, collect the items listed in “Diagnostics” on page 1103 and contact IBM Support.

00F3021E

An ESTAE could not be established while in the process of routing control to the actual ESTAE routine. The caller (RTM) is ended abnormally. This causes the original error to percolate to a higher-level recovery routine and causes this reason code to be shown in an RTM recovery environment.

This can occur if the current address space (usually an allied address space) has insufficient storage.

System action

The caller is ended abnormally and a dump is produced.

System programmer response

Examine the usage and free areas in the LSQA portion of the current address space private area. If necessary, have the size of the private area expanded.

You might find the items listed in “Diagnostics” on page 1103 useful in resolving the problem.

00F3021F, 00F30220

An internal error has occurred.

System action

The caller is not ended abnormally. A dump is taken, and an entry is written in SYS1.LOGREC.

System programmer response

Collect the items listed in “Diagnostics” on page 1103 and contact IBM Support.

00F30230

An internal error has occurred.

System action

The connection between the allied address space and the queue manager terminated. A dump is taken, and an entry is written in SYS1.LOGREC.

System programmer response

Collect the items listed in “Diagnostics” on page 1103 and contact IBM Support.

00F30310

An internal error has occurred.

System action

The invoker is ended abnormally. A dump is taken, and an entry is written in SYS1.LOGREC.

System programmer response

Collect the items listed in “Diagnostics” on page 1103 and contact IBM Support.

00F30311

An ESTAE could not be established during the processing of a resolve-indoubt request. This can occur if the current address space has insufficient storage. This will probably cause an abnormal termination of the queue manager.

System action

The caller is ended abnormally.

System programmer response

Restart the queue manager if necessary.

Examine the usage and free areas in the local system queue area (LSQA) portion of the current address space private area. If necessary, have the size of the private area expanded.

The caller should produce a SYS1.LOGREC entry and an SVC dump, so that you can examine the LSQA area.

You might find the items listed in "Diagnostics" on page 1103 useful in resolving the problem.

00F30312

An ESTAE could not be established during the processing of a resolve-indoubt-UR request. This can occur if the current address space has insufficient storage.

System action

The caller is ended abnormally.

System programmer response

Examine the usage and free areas in the local system queue area (LSQA) portion of the current address space private area. If necessary, have the size of the private area expanded.

The caller should produce a SYS1.LOGREC entry and an SVC dump.

You might find the items listed in "Diagnostics" on page 1103 useful in resolving the problem.

00F30313

A control block could not be allocated. This could occur when the storage pool has no more free space available.

System action

The request is not processed. The application program is ended abnormally with completion code X'5C6' and this reason code.

System programmer response

A dump should be taken for problem analysis.

Check that you are running with the recommended region size, and if not, reset your system and retry. If you are unable to resolve the problem, collect the items listed in "Diagnostics" on page 1103 and contact IBM Support.

00F30400, 00F30401, 00F30402

An internal error has occurred.

System action

The program which made the request might produce diagnostics to report the error.

System programmer response

Collect the diagnostics produced by the application program reporting the error, if any, and contact IBM Support.

00F30406

The queue manager has gone to EOM (end-of-memory). This is probably because the z/OS command **FORCE** has been issued.

System action

The queue manager is terminated, and a dump is taken.

System programmer response

The queue manager can be restarted after termination completes.

Determine why the z/OS command **FORCE** was issued.

00F30409, 00F3040A

An internal error has occurred.

System action

The queue manager is terminated with an SVC dump.

System programmer response

The queue manager can be started again after it terminates.

Collect the items listed in “Diagnostics” on page 1103 and contact IBM Support.

00F3040B

See message CSQ3001E.

System action

See message CSQ3001E.

System programmer response

See message CSQ3001E.

You might find the items listed in “Diagnostics” on page 1103 useful in resolving the problem.

00F3040C, 00F3040D

An internal error has occurred.

System action

The queue manager is terminated with an SVC dump.

System programmer response

The queue manager can be started again after it terminates.

Collect the items listed in “Diagnostics” on page 1103 and contact IBM Support.

00F3040E

An internal error has occurred.

System action

The queue manager is terminated.

System programmer response

The queue manager should be restarted.

Collect the items listed in “Diagnostics” on page 1103 and contact IBM Support.

00F3040F, 00F30410

An internal error has occurred.

System action

The queue manager is terminated.

System programmer response

The queue manager can be started again after it terminates.

Collect the items listed in “Diagnostics” on page 1103 and contact IBM Support.

00F30411, 00F30412, 00F30413

An internal error has occurred.

System action

The queue manager is terminated.

System programmer response

The queue manager can be started again after it terminates.

Collect the items listed in “Diagnostics” on page 1103 and contact IBM Support.

00F30414

An internal error has occurred.

System action

The queue manager is terminated.

System programmer response

The queue manager can be started again after it terminates. If the problem persists, request a stand-alone dump, and perform an IPL of your z/OS system.

Collect the items listed in “Diagnostics” on page 1103 and contact IBM Support.

00F30415

An ESTAE could not be established during the processing of an EOM SSI broadcast. This is probably a z/OS problem, because these modules are executing in the z/OS master scheduler address space.

System action

The queue manager is terminated.

System programmer response

The queue manager can be started again after it terminates. If the problem persists, it might be necessary to perform an IPL of your z/OS system.

This can occur if the z/OS master scheduler address space has insufficient free storage. If such is the case, IBM MQ is unable to write a SYS1.LOGREC record or request a dump. The z/OS master scheduler should have produced these diagnostic aids. Examine the dump to determine whether the problem is in z/OS or IBM MQ. Other unrelated errors in the z/OS Master Scheduler address space would indicate a z/OS problem.

If the problem appears to be an IBM MQ problem, collect the items listed in “Diagnostics” on page 1103 and contact IBM Support.

00F30416

An ESTAE could not be established during the processing of an EOM for an allied address space.

System action

The queue manager is terminated.

System programmer response

The queue manager can be started again after it terminates. If the problem persists, it might be necessary to perform an IPL of your z/OS system.

This can occur if the z/OS master scheduler address space has insufficient free storage. If such is the case, IBM MQ is unable to write a SYS1.LOGREC record or request a dump. The z/OS master scheduler should have produced these diagnostic aids. Examine the dump to determine whether the problem is in z/OS or IBM MQ. Other unrelated errors in the z/OS Master Scheduler address space would indicate a z/OS problem.

If the problem appears to be an IBM MQ problem, collect the items listed in “Diagnostics” on page 1103 and contact IBM Support.

00F30417, 00F30418

An internal error has occurred.

System action

The queue manager is terminated.

System programmer response

The queue manager can be started again after it terminates.

Collect the items listed in "Diagnostics" on page 1103 and contact IBM Support.

00F30419

An internal error has occurred.

System action

The queue manager is terminated with an SVC dump.

System programmer response

The queue manager can be started again after it terminates.

Collect the items listed in "Diagnostics" on page 1103 and contact IBM Support.

00F3041A

An ESTAE could not be established by the deferred end-of-task (EOT) processor. This error could occur only during queue manager startup. Probably, an ESTAE could not be established because of a shortage of LSQA space.

System action

The queue manager is terminated.

System programmer response

Restart the queue manager.

If the problem persists, increase the size of the queue manager address space private area.

You might find the items listed in "Diagnostics" on page 1103 useful in resolving the problem.

00F3041B, 00F30420

An internal error has occurred.

System action

The queue manager is terminated. A SYS1.LOGREC entry and associated SVC dump were requested.

System programmer response

Restart the queue manager.

Collect the items listed in "Diagnostics" on page 1103 and contact IBM Support.

00F30429

An internal error has occurred.

System action

The queue manager is terminated with an SVC dump.

System programmer response

Restart the queue manager.

Collect the items listed in "Diagnostics" on page 1103 and contact IBM Support.

00F30450

An ESTAE could not be established during the processing of an identify SSI call. This can occur if the current address space has insufficient storage.

System action

The allied address space is ended abnormally (without a dump). A dump should be produced by the allied task.

System programmer response

The user can retry the identify request. If a dump is available, review the storage manager's control blocks to determine if all of the private area has been allocated. If necessary, increase the private area size of the allied address space.

You might find the items listed in "Diagnostics" on page 1103 useful in resolving the problem.

00F30451

An ESTAE could not be established during the processing of an identify SSI call. This can occur if the current address space has insufficient storage.

System action

The allied task is ended abnormally (without a dump). A dump should be produced by the allied task.

System programmer response

The user can retry the identify request. If a dump is available, review the storage manager's control blocks to determine if all of the private area has been allocated. If necessary, increase the private area size of the allied address space.

You might find the items listed in "Diagnostics" on page 1103 useful in resolving the problem.

00F30452

An ESTAE could not be established during the processing of an identify SSI call. This can occur if the current address space has insufficient storage.

System action

The allied task is ended abnormally (without a dump). A dump should be produced by the allied task.

System programmer response

The user can retry the identify request. If a dump is available, review the storage manager's control blocks to determine if all of the private area has been allocated. If necessary, increase the private area size of the allied address space.

You might find the items listed in "Diagnostics" on page 1103 useful in resolving the problem.

00F30453

ESTAEs could not be established during the processing of a n SSI call other than FEOT, EOM, HELP, COMMAND, and IDENTIFY. This can occur if the current address space has insufficient storage.

System action

The allied task is ended abnormally (without a dump). A dump should be produced by the allied task.

System programmer response

The user can retry the request. If a dump is available, review the storage manager's control blocks to determine if all of the private area has been allocated. If necessary, increase the private area size of the allied address space.

You might find the items listed in "Diagnostics" on page 1103 useful in resolving the problem.

00F30454

An internal error has occurred.

System action

The allied task is ended abnormally.

System programmer response

Collect the items listed in “Diagnostics” on page 1103 and contact IBM Support.

00F30455

An ESTAE could not be established during the processing of an identify termination request. This can occur if the current address space has insufficient storage.

System action

The allied task is ended abnormally (without a dump). A dump should be produced by the allied task.

System programmer response

The user can retry the request. If a dump is available, review the storage manager's control blocks to determine if all of the private area has been allocated. If necessary, increase the private area size of the allied address space.

You might find the items listed in “Diagnostics” on page 1103 useful in resolving the problem.

00F30456

An internal error has occurred.

System action

The calling task is ended abnormally.

System programmer response

Collect the items listed in “Diagnostics” on page 1103 and contact IBM Support.

00F30457

An internal error has occurred.

System action

The caller is ended abnormally. The error might, in many cases, eventually terminate the queue manager.

System programmer response

Restart the queue manager if necessary.

Collect the items listed in “Diagnostics” on page 1103 and contact IBM Support.

00F30459

An internal error has occurred.

System action

The queue manager is terminated with a reason code of X'00F30420'.

System programmer response

Restart the queue manager.

Collect the items listed in “Diagnostics” on page 1103 and contact IBM Support.

00F30461

The queue manager was unable to successfully restart with RRS because of an internal error in either IBM MQ or RRS.

System action

The queue manager is not connected to RRS and all services dependent on that connection are unavailable. This means that applications might not connect to the queue manager using RRSAF and that WLM-established address spaces might not be used for IBM MQ stored procedures until the queue manager successfully restarts with RRS.

System programmer response

Stop and then start RRS. Stop and then start the queue manager. If the problem persists, perform an RRS cold start.

Collect the items listed in "Diagnostics" on page 1103 and contact IBM Support.

00F30501, 00F30502

An internal error has occurred.

System action

The requester is ended abnormally, and the request is not processed.

System programmer response

Collect the items listed in "Diagnostics" on page 1103 and contact IBM Support.

00F30503

CSQ6SYSP is missing from the system parameter load module.

System action

Queue manager start-up is terminated.

System programmer response

Re-create your system parameter load module (if a customized version is being used) and restart the queue manager. For information about the system parameter modules, see Tailor your system parameter module.

00F30573, 00F30574

An internal error has occurred.

System action

The requester is ended abnormally, and the request is not processed. A dump is taken, and an entry is written in SYS1.LOGREC.

System programmer response

Collect the items listed in "Diagnostics" on page 1103 and contact IBM Support.

00F30580

An internal error has occurred.

System action

The requester is ended abnormally.

System programmer response

Collect the items listed in "Diagnostics" on page 1103 and contact IBM Support.

00F30581

An internal error has occurred.

System action

The queue manager ends abnormally. The startup/shutdown ESTAE creates a SYS1.LOGREC entry and takes an SVC dump.

System programmer response

Restart the queue manager.

Collect the items listed in "Diagnostics" on page 1103 and contact IBM Support.

00F30597, 00F30598

An internal error has occurred.

System action

The allied task is ended abnormally, and the request is not processed.

System programmer response

Collect the items listed in “Diagnostics” on page 1103 and contact IBM Support.

00F30599

An internal error has occurred.

System action

The connection name associated with the error is probably unable to continue communication with IBM MQ until the queue manager is terminated and restarted.

System programmer response

If necessary, stop and restart the queue manager.

Collect the items listed in “Diagnostics” on page 1103 and contact IBM Support.

00F30601

Asynchronous events occurred which caused the premature termination of the thread. The thread could not be recovered.

There might be other errors or messages concerning this allied user indicating what the asynchronous events were.

System action

The allied user is ended abnormally with completion code X'5C6' and this reason code.

System programmer response

You might find the items listed in “Diagnostics” on page 1103 useful in resolving the problem.

00F30610

An ESTAE could not be established during the processing of an 'end stop-work force' notification. This can occur if there is insufficient storage. This might lead to abnormal termination of the queue manager.

System action

The caller is ended abnormally. An SVC dump and related SYS1.LOGREC entry are requested.

System programmer response

If necessary, restart the queue manager.

If necessary, increase the private area size of the address space.

You might find the items listed in “Diagnostics” on page 1103 useful in resolving the problem.

00F30801

An internal error has occurred.

System action

The queue manager is terminated. An SVC dump is requested.

System programmer response

Restart the queue manager.

Collect the items listed in “Diagnostics” on page 1103 and contact IBM Support.

00F30802

An internal error has occurred.

System action

The task is not ended abnormally.

System programmer response

Collect the items listed in “Diagnostics” on page 1103 and contact IBM Support.

00F30803

An ESTAE could not be established during the processing of an application program support call. This can occur if the current address space has insufficient storage.

System action

The allied task is ended abnormally. The allied task might have requested an SVC dump.

System programmer response

The user can retry the request. If necessary, increase the private area size of the application address space.

You might find the items listed in “Diagnostics” on page 1103 useful in resolving the problem.

00F30805

An internal error has occurred.

System action

The request might have been processed or rejected.

System programmer response

Collect the items listed in “Diagnostics” on page 1103 and contact IBM Support.

00F30901

IBM MQ has lost its cross-memory authority to an allied address space because the ally has released its authorization index.

System action

The allied address space is terminated.

System programmer response

You might find the items listed in “Diagnostics” on page 1103 useful in resolving the problem.

00F30902

MQ has detected a recursive error condition while processing End-of-Task for a task in an allied address space.

System action

The allied address space is terminated.

System programmer response

You might find the items listed in “Diagnostics” on page 1103 useful in resolving the problem.

00F30903

An error has occurred while processing End-of-Task for the queue manager address space.

System action

The address space is forced to 'end-of-memory' with this reason code.

System programmer response

You might find the items listed in “Diagnostics” on page 1103 useful in resolving the problem.

00F30904

End-of-Task occurred for the queue manager address space, and IBM MQ could not establish an ESTAE to protect its processing. Insufficient storage might be the reason the ESTAE could not be established.

System action

The address space is forced to 'end-of-memory' with this reason code.

System programmer response

You might find the items listed in “Diagnostics” on page 1103 useful in resolving the problem.

Attempt to determine if one or more IBM MQ address spaces is storage-constrained. Examination of the console output for the time period preceding this condition might reveal other messages or indications that the terminating address space was storage-constrained.

00F30905

End-of-Task occurred for the job step task in an allied address space. IBM MQ would normally attempt to terminate the address space's connection to the queue manager but was unable to protect its processing by establishing an ESTAE. Insufficient storage might be the reason the ESTAE could not be established.

System action

The address space is forced to 'end-of-memory' with this reason code.

System programmer response

You might find the items listed in “Diagnostics” on page 1103 useful in resolving the problem.

Attempt to determine if one or more allied address spaces is storage-constrained. Examination of the console output for the time period preceding this condition might reveal other messages or indications that the terminating allied address space was storage-constrained.

00F33100

The IBM MQ thread is read-only.

System action

A prepare issued by the application program was processed through Phase-1. IBM MQ discovered there were no resources modified and no need for COMMIT or BACKOUT to be subsequently issued.

System programmer response

This might create a path length saving by not issuing the subsequent commit or backout which normally follows prepare. No further action is required to complete the unit of recovery; the unit of recovery is complete.

Related tasks

Collecting troubleshooting information

z/OS *Db2 manager codes (X'F5')*

If a Db2 manager reason code occurs that is not listed here, an internal error has occurred. Collect the following diagnostic items and contact IBM Support.

Diagnostics

- A description of the actions that led to the error or, if applicable, either a listing of the application program or the input string to a utility program that was being run at the time of the error.

- The console output for the period leading up to the error.
- The queue manager job log.
- The system dump resulting from the error.
- A printout of SYS1.LOGREC.
- Appropriate IBM MQ, z/OS, Db2, CICS, and IMS service levels.

00F50000

An internal error has occurred.

System action

The queue manager terminates, a record is written to SYS1.LOGREC and a dump is taken.

System programmer response

Ensure that the QSGDATA system parameter is specified correctly and restart the queue manager.

If the problem persists, collect the items listed in "Diagnostics" on page 1120 and contact IBM Support.

00F50001

An internal error has occurred.

System action

The queue manager terminates, a record is written to SYS1.LOGREC and a dump is taken.

System programmer response

Restart the queue manager.

If the problem persists, collect the items listed in "Diagnostics" on page 1120 and contact IBM Support.

00F50002

An internal error has occurred.

System action

The task ends abnormally. Queue manager processing continues but the queue manager might not terminate normally and might not register Db2 termination.

System programmer response

Refer to *Db2 for z/OS Messages and Codes* for information about the completion and reason code in the accompanying message and collect the diagnostic data requested in the manual. In addition, collect the items listed in "Diagnostics" on page 1120 and contact IBM Support.

00F50003

An internal error has occurred.

System action

The task ends abnormally. Queue manager processing continues.

System programmer response

Collect the items listed in "Diagnostics" on page 1120 and contact IBM Support.

00F50004

An internal error has occurred.

System action

The queue manager terminates, a record is written to SYS1.LOGREC and a dump is taken.

System programmer response

Ensure that the following modules are available through the linklist or the steplib concatenation: DSNRLI, DSNHLIR, DSNWLIR, ATRCMIT and ATRBACK. Restart the queue manager.

If the problem persists, collect the items listed in "Diagnostics" on page 1120 and contact IBM Support.

00F50006

An internal error has occurred.

System action

The queue manager terminates, a record is written to SYS1.LOGREC and a dump is taken.

System programmer response

All queue managers that are members of the same queue sharing group must connect to the same Db2 data-sharing group. Check that all queue managers in the queue sharing group have the same Db2 data-sharing group specified in the QSGDATA system parameter. Restart the queue manager.

Collect the items listed in "Diagnostics" on page 1120 and contact IBM Support.

00F50007

An internal error has occurred.

System action

The queue manager terminates, a record is written to SYS1.LOGREC and a dump is taken.

System programmer response

Ensure that the Db2 subsystem(s) specified on the QSGDATA system parameter are members of the Db2 data-sharing group that is also specified on the QSGDATA system parameter. Restart the queue manager.

If the problem persists, refer to *Db2 for z/OS Messages and Codes* for information about the completion and reason code in the accompanying message and collect the diagnostic data requested in the manual. In addition, collect the items listed in "Diagnostics" on page 1120 and contact IBM Support.

00F50008

An internal error has occurred.

System action

The task ends abnormally and processing continues.

System programmer response

Collect the items listed in "Diagnostics" on page 1120 and contact IBM Support.

00F50009

An internal error has occurred.

System action

The queue manager terminates, a record is written to SYS1.LOGREC and a dump is taken.

System programmer response

Restart the queue manager.

Refer to *Db2 for z/OS Messages and Codes* for information about the completion and reason code in the accompanying message and collect the diagnostic data requested in the manual. In addition, collect the items listed in "Diagnostics" on page 1120 and contact IBM Support.

00F50010

An internal error has occurred.

System action

The queue manager terminates, a record is written to SYS1.LOGREC and a dump is taken.

System programmer response

Restart the queue manager.

See z/OS MVS Programming: Sysplex Services Reference for an explanation of the error and the diagnostic information, if any, that you must collect. In addition, collect the items listed in "Diagnostics" on page 1120 and contact IBM Support.

00F50013

No queue manager entry was found in the CSQ.ADMIN_B_QMGR table for this combination of queue manager and queue sharing group, or the entry was incorrect.

System action

The queue manager terminates, a record is written to SYS1.LOGREC and a dump is taken.

System programmer response

Check the CSQ.ADMIN_B_QMGR table in the Db2 data-sharing group and ensure that an entry has been defined for the queue manager and it relates to the correct queue sharing group.

If you are migrating from a previous release of MQ, check also that you have updated the Db2 tables to the format for the current release. See Maintaining and migrating, for information about migration and compatibility between releases.

Restart the queue manager. If the problem persists, collect the items listed in "Diagnostics" on page 1120 and contact IBM Support.

00F50014

An internal error has occurred.

System action

The queue manager terminates, a record is written to SYS1.LOGREC and a dump is taken.

System programmer response

Check that the Db2 related installation and customization tasks have all completed successfully. Restart the queue manager.

If the problem persists, refer to *Db2 for z/OS Messages and Codes* for information about the completion and reason code in the accompanying message and collect the diagnostic data requested in the manual. In addition, collect the items listed in "Diagnostics" on page 1120 and contact IBM Support.

00F50015

An internal error has occurred.

System action

The queue manager terminates, a record is written to SYS1.LOGREC and a dump is taken.

System programmer response

Restart the queue manager.

If the problem persists, refer to *Db2 for z/OS Messages and Codes* for information about the completion and reason code in the accompanying message and collect the diagnostic data requested in the manual. In addition, collect the items listed in "Diagnostics" on page 1120 and contact IBM Support.

00F50016

An internal error has occurred.

System action

The queue manager terminates, a record is written to SYS1.LOGREC and a dump is taken.

System programmer response

Restart the queue manager.

If the problem persists, refer to *Db2 for z/OS Messages and Codes* for information about the completion and reason code in the accompanying message and collect the diagnostic data requested in the manual. In addition, collect the items listed in "Diagnostics" on page 1120 and contact IBM Support.

00F50017

An internal error has occurred.

System action

The queue manager terminates, a record is written to SYS1.LOGREC and a dump is taken.

System programmer response

See z/OS MVS Programming: Sysplex Services Reference for information about the completion and reason code in the accompanying message.

Restart the queue manager. If the problem persists, collect the items listed in "Diagnostics" on page 1120 and contact IBM Support.

This error may occur if one or more of the queue managers in a queue sharing group (QSG) do not have a member entry in the XCF group for the QSG.

Enter the following z/OS command substituting the queue sharing group name for xxxx:

```
D XCF,GRP,CSQGxxxx,ALL
```

This will list the members of the XCF group. If any queue managers are defined as a member of the QSG, but do not have an entry in the XCF Group, use the ADD QMGR command of the CSQ5PQSG utility to restore the XCF group entry for that queue manager. The utility should be run for each queue manager which does not have an entry in the XCF group.

00F50018

An internal error has occurred.

System action

The queue manager terminates, a record is written to SYS1.LOGREC and a dump is taken.

System programmer response

See z/OS MVS Programming: Sysplex Services Reference for information about the completion and reason code in the accompanying message.

Restart the queue manager. If the problem persists, collect the items listed in "Diagnostics" on page 1120 and contact IBM Support.

00F50019

An internal error has occurred.

System action

The queue manager terminates, a record is written to SYS1.LOGREC and a dump is taken.

System programmer response

See z/OS MVS Programming: Sysplex Services Reference for information about the completion and reason code in the accompanying message.

Restart the queue manager. If the problem persists, collect the items listed in "Diagnostics" on page 1120 and contact IBM Support.

00F5001C

CSQ5_DB2_UNAVAILABLE

System action

The queue manager terminates, a record is written to SYS1.LOGREC and a dump is taken.

System programmer response

See z/OS MVS Programming: Sysplex Services Reference for information about the completion and reason code in the accompanying message.

Restart the queue manager. If the problem persists, collect the items listed in "Diagnostics" on page 1120 and contact IBM Support.

00F50021

An internal error has occurred.

System action

The queue manager terminates, a record is written to SYS1.LOGREC and a dump is taken.

System programmer response

See z/OS MVS Programming: Sysplex Services Reference for information about the completion and reason code in the accompanying message.

Restart the queue manager. If the problem persists, collect the items listed in "Diagnostics" on page 1120 and contact IBM Support.

00F50024

An internal error has occurred.

System action

The task ends abnormally and a dump is taken.

System programmer response

If the problem persists, collect the items listed in "Diagnostics" on page 1120 and contact IBM Support.

00F50025

An internal error has occurred.

System action

The task ends abnormally and a dump is taken.

System programmer response

Collect the items listed in "Diagnostics" on page 1120 and contact IBM Support.

00F50026

An internal error has occurred.

System action

The task ends abnormally and a dump is taken.

System programmer response

Collect the items listed in “Diagnostics” on page 1120 and contact IBM Support.

00F50027

An internal error has occurred.

System action

The task ends abnormally and a dump is taken.

System programmer response

Collect the items listed in “Diagnostics” on page 1120 and contact IBM Support.

00F50028

An internal error has occurred.

System action

The task ends abnormally and a dump is taken.

System programmer response

This might be a temporary condition if Db2 or RRS has failed. If the problem persists, collect the items listed in “Diagnostics” on page 1120, together with output from Db2 command **DISPLAY THREAD(*)**, and contact IBM Support.

00F50029

The queue manager has detected a mismatch between its supported versions of MQ and those of other members of the queue sharing group.

System action

The queue manager terminates, a record is written to SYS1.LOGREC and a dump is taken.

System programmer response

Locate the queue managers in the queue-sharing group that are below the level supported by the queue manager issuing the message, and upgrade them to at least the supported level before trying to start the queue manager again.

For information about compatibility between releases, see Queue sharing group coexistence on z/OS.

If the correct version is being run, collect the items listed in Db2, together with a printout of the CSQ.ADMIN_B_QMGR table from the Db2 data-sharing group to which the queue manager connected, and contact IBM Support.

00F50033

The queue manager detected that one or more active log datasets is encrypted, and active log encryption is not supported by one or more other members of the queue sharing group.

This abend is preceded by message CSQ5040E. See the explanation of that message for more details of the failure

System action

The queue manager terminates, a record is written to SYS1.LOGREC and a dump is taken.

00F50901

An internal error has occurred.

System action

The job ends abnormally with a X'5C6' completion code and a dump is taken.

System programmer response

Collect the items listed in “Diagnostics” on page 1120 and contact IBM Support.

00F51030

An internal error has occurred.

System action

The task ends abnormally and a dump is taken.

System programmer response

Restart RRS if it has terminated. If RRS has not terminated, collect the items listed in “Diagnostics” on page 1120 and contact IBM Support.

00F51031

An internal error has occurred on a Db2 connection thread.

System action

The task ends abnormally and a new task is created. A dump is taken if there is an 'in-flight' Db2 request.

System programmer response

None. A new Db2 server task is automatically re-created to replace the task that was terminated. If the problem persists, collect the items listed in “Diagnostics” on page 1120 and contact IBM Support.

Related tasks

Collecting troubleshooting information

z/OS *Generalized command preprocessor codes (X'F9')*

If a command preprocessor reason code occurs that is not listed here, an internal error has occurred. Collect the following diagnostic items and contact IBM Support.

Diagnostics

- A description of the actions that led to the error or, if applicable, either a listing of the application program or the input string to a utility program that was being run at the time of the error.
- The console output for the period leading up to the error.
- The queue manager job log.
- The system dump resulting from the error.
- Appropriate IBM MQ, z/OS, Db2, CICS, and IMS service levels.
- If you are using the IBM MQ Operations and Control panels, the ISPF panel name.
- The command issued before the error occurred.

00F90000

An internal error has occurred.

System action

Command execution was ended abnormally. If the command was properly entered, it might have been partially or completely executed.

System programmer response

Collect the items listed in “Diagnostics” on page 1127 and contact IBM Support.

It might be necessary to restart the CICS or IMS adapter.

00F90001

An internal error has occurred.

System action

Command execution was ended abnormally. If the command was properly entered, it might have been partially or completely executed.

System programmer response

Collect the items listed in "Diagnostics" on page 1127 and contact IBM Support.

It might be necessary to restart the CICS or IMS adapter.

00F90002

The routines of the multiple console support (MCS) service of z/OS. were unable to initialize. This condition might indicate an error in the address space.

System action

Initialization is stopped, causing the queue manager to terminate.

System programmer response

Collect the items listed in "Diagnostics" on page 1127 and contact IBM Support.

Restart the queue manager.

00F90003

The routines of the multiple console support (MCS) service of z/OS were unable to initialize.

System action

If the error was issued by module CSQ9SCNM, queue manager initialization is stopped, causing the queue manager to terminate. If the error was issued by module CSQ9SCN6, the command from the associated console is executed, and should proceed normally.

System programmer response

Collect the items listed in "Diagnostics" on page 1127 and contact IBM Support.

00F90004

The routines of the multiple console support (MCS) service of z/OS detected a logic error.

System action

The command was not executed.

System programmer response

Collect the items listed in "Diagnostics" on page 1127 and contact IBM Support.

00F90005

A routine of the multiple console support (MCS) service of z/OS was not able to create an ESTAE recovery environment. This condition is detected when the ESTAE service of z/OS returns a nonzero return code. The command from the associated z/OS console is not executed. See *z/OS MVS Programming: Assembler Services Reference* Volume 1 for an explanation of ESTAE return codes.

System action

Command processing is terminated.

System programmer response

Collect the items listed in "Diagnostics" on page 1127 and contact IBM Support.

00F90006

An internal error has occurred.

System action

Agent allocation is terminated.

System programmer response

Collect the items listed in “Diagnostics” on page 1127 and contact IBM Support.

00F90007

An internal error has occurred.

System action

The statistical update is not completed. The statistics block address is cleared from the CGDA to prevent future problems. No further command statistical counts are maintained. Processing for the command is retried and should complete normally.

System programmer response

Collect the items listed in “Diagnostics” on page 1127 and contact IBM Support.

00F90008

An internal error has occurred.

System action

The function is ended abnormally.

System programmer response

Collect the items listed in “Diagnostics” on page 1127 and contact IBM Support.

00F90009

This reason code is used to document that module CSQ9SCN9 has added information to the SDWA variable recording area (VRA) following the data provided by the CSQWRCRD service. If CSQ9SCN9 records an error in SYS1.LOGREC and the reason code in the VRA is not of the form X'00F9xxxx', the reason code is changed to X'00F90009'. This is done so that anyone examining a SYS1.LOGREC entry can determine, from the reason code, what additional data has been placed in the VRA. The reason code is the first data item in the VRA, as mapped by macro IHAVRA.

System programmer response

Collect the items listed in “Diagnostics” on page 1127 and contact IBM Support.

00F9000A

An internal error has occurred.

System action

Command execution was ended abnormally. The command was not executed.

System programmer response

Collect the items listed in “Diagnostics” on page 1127 and contact IBM Support.

00F9000B

An internal error occurred while attempting to obtain CSA storage. The storage request could not be satisfied, either because no CSA storage was available or because an unreasonably large amount of storage was requested. The amount of storage requested is determined by the length of the command being parsed. Normally, it is several hundred bytes.

System action

Command execution is ended abnormally.

System programmer response

It might be necessary to restart the CICS or IMS adapter, or the queue manager.

If the problem persists, collect the items listed in “Diagnostics” on page 1127 and contact IBM Support.

00F9000C

An internal error has occurred.

The command processor invoked attempted to return a message formatted for inclusion in a z/OS multiple line WTO (write to operator).

System action

Command execution is ended abnormally.

System programmer response

The command in error is identified by message CSQ9017E. It might be necessary to restart the CICS or IMS adapter, or the queue manager.

Collect the items listed in "Diagnostics" on page 1127 and contact IBM Support.

00F9000D

An internal error has occurred.

System action

The queue manager start-up is terminated.

System programmer response

Restart the queue manager.

Collect the items listed in "Diagnostics" on page 1127 and contact IBM Support.

00F9000E

An internal error has occurred.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Collect the items listed in "Diagnostics" on page 1127 and contact IBM Support.

00F9000F

MQ was unable to locate the default userid to be used on a command check. This indicates that CSQ6SYSP is not in the system parameter load module.

System action

The current execution unit terminates with completion code X'5C6'.

System programmer response

Ensure that CSQ6SYSP is in the system parameter load module. Restart the queue manager if necessary.

00F90010

An internal error has occurred while processing a command.

System action

Command execution was ended abnormally. The command was not executed.

System programmer response

Collect the items listed in "Diagnostics" on page 1127 and contact IBM Support.

Related tasks

Collecting troubleshooting information

z/OS IBM MQ CICS adapter abend codes

All the CICS versions supported by IBM MQ 9.0.0, and later, use the CICS supplied version of the adapter. See the Transaction abend codes section of the CICS documentation for further information.

z/OS IBM MQ CICS bridge abend codes

All the CICS versions supported by IBM MQ 9.0.0, and later, use the CICS supplied version of the bridge. See the Transaction abend codes section of the CICS documentation for further information.

z/OS IBM MQ component identifiers

IBM MQ for z/OS has a component-based architecture and each component uses a unique identifier code. These identifier codes are displayed in some of the informational messages.

Table 16. Component identifiers used in IBM MQ messages and codes

Component	ID	Hex ID
Batch adapter	B	X'C2'
CICS adapter	C	X'C3'
Coupling Facility manager	E	X'C5'
Message generator	F	X'C6'
Functional recovery manager	G	X'C7'
Security manager	H	X'C8'
Data manager	I	X'C9'
Recovery log manager	J	X'D1'
Lock manager	L	X'D3'
Connection manager	m	X'94'
Message manager	M	X'D4'
Command server	N	X'D5'
Operations and control	O	X'D6'
Buffer manager	P	X'D7'
IMS adapter	Q	X'D8'
Recovery manager	R	X'D9'
Storage manager	S	X'E2'
Timer services	T	X'E3'
Utilities	U	X'E4'
Agent services	V	X'E5'
Instrumentation facilities	W	X'E6'
Distributed queuing	X	X'E7'
Initialization procedures and general services	Y	X'E8'
System parameter manager	Z	X'E9'

Table 16. Component identifiers used in IBM MQ messages and codes (continued)		
Component	**ID**	**Hex ID**
Advanced message security	0 (zero)	X'F0'
Service facilities	1	X'F1'
IBM MQ - IMS bridge	2	X'F2'
Subsystem support	3	X'F3'
Db2 manager	5	X'F5'
Generalized command processor	9	X'F9'

z/OS Communications protocol return codes for z/OS

The communication protocols used by IBM MQ for z/OS can issue their own return codes. Use these tables to identify the return codes used by each protocol.

The tables in this topic show the common return codes from TCP/IP and APPC/MVS returned in messages from the distributed queuing component:

- "TCP/IP z/OS UNIX System Services Sockets return codes" on page 1132
- APPC/MVS return codes

If the return code is not listed, or if you want more information, see to the documentation mentioned in each table.

If the return code you received is X'7D0' or more, it is one of the MQRC_* return codes issued by IBM MQ. These codes are listed in "API completion and reason codes" on page 1151.

TCP/IP z/OS UNIX System Services Sockets return codes

See the *z/OS UNIX System Services Messages and Codes* manual for more information and for further return codes.

Table 17. z/OS UNIX System Services sockets return codes	
Return code (Hexadecimal)	**Explanation**
0001	Error in the domain
0002	Result is too large
006F	Permission is denied
0070	The resource is temporarily unavailable
0071	The file descriptor is incorrect
0072	The resource is busy
0073	No child process exists
0074	A resource deadlock is avoided
0075	The file exists
0076	The address is incorrect
0077	The file is too large
0078	A function call is interrupted

Table 17. z/OS UNIX System Services sockets return codes (continued)	
Return code (Hexadecimal)	**Explanation**
0079	The parameter is incorrect
007A	An I/O error occurred
007B	The file specified is a directory
007C	Too many files are open for this process
007D	Too many links occurred
007E	The file name is too long
007F	Too many files are open in the system
0080	No such device exists
0081	No such file, directory, or IPC member exists
0082	The exec call contained a format error (DFSMS error)
0083	No locks are available
0084	Not enough space is available
0085	No space is left on the device, or no space is available to create the IPC member ID
0086	The function is not implemented
0087	Not a directory
0088	The directory is not empty
0089	The I/O control operator is inappropriate
008A	No such device or address exists
008B	The operation is not permitted
008C	The pipe is broken
008D	The specified file system is read only
008E	The seek is incorrect
008F	No such process or thread exists
0090	A link to a file on another file system was attempted
0091	The parameter list is too long, or the message to receive was too large for the buffer
0092	A loop is encountered in symbolic links
0093	The byte sequence is incorrect
0095	A value is too large to be stored in the data type
0096	OpenMVS kernel is not active
0097	Dynamic allocation error
0098	Catalog Volume Access Facility error
0099	Catalog obtain error

Table 17. z/OS UNIX System Services sockets return codes (continued)	
Return code (Hexadecimal)	**Explanation**
009C	Process Initialization error
009D	An MVS environmental or internal error has occurred
009E	Bad parameters were passed to the service
009F	File system encountered a permanent file error
00A2	File system encountered a system error
00A3	SAF/RACF extract error
00A4	SAF/RACF error
00A7	Access to the OpenMVS version of the C RTL is denied
00A8	The password for the specified resource has expired
00A9	The new password specified is not valid
00AA	A WLM service ended in error
03EA	Socket number assigned by client interface code (for socket() and accept()) is out of range
03EB	Socket number assigned by client interface code is already in use
03ED	Offload box error
03EE	Offload box restarted
03EF	Offload box down
03F0	Already a conflicting call outstanding on socket
03F1	Request canceled using SOCKcallCANCEL request
03F3	SetIbmOpt specified a name of a PFS that either was not configured or was not a Sockets PFS
044C	Block device required
044D	Text file busy
044E	The descriptor is marked nonblocking, and the requested function cannot complete immediately
044F	Operation now in progress
0450	Operation already in progress
0451	Socket operation on a non-socket
0452	Destination address required
0453	The message is too large to be sent in a single transmission, as required
0454	The socket type is incorrect
0455	Protocol or socket option unavailable
0456	Protocol not supported
0457	Socket type not supported
0458	The referenced socket is not a type that supports the requested function

Table 17. z/OS UNIX System Services sockets return codes (continued)	
Return code (Hexadecimal)	**Explanation**
0459	Protocol family not supported
045A	The address family is not supported
045B	The address is already in use
045C	Cannot assign requested address
045D	Network is down
045E	Network is unreachable
045F	Network dropped connection on reset
0460	Software caused connection abort
0461	Connection reset by peer
0462	Insufficient buffer space available
0463	The socket is already connected
0464	The socket is not connected
0465	Cannot send after socket shutdown
0466	Too many references: Cannot splice
0467	Connection timed out
0468	The attempt to connect was rejected
0469	Host is down
046A	No route to host
046B	Too many processes
046C	Too many users
046D	Disk quota exceeded
046E	Stale NFS file handle
046F	Too many levels of remote in path
0470	Device is not a stream
0471	Timer expired
0472	Out of streams resources
0473	No message of the required type
0474	Trying to read unreadable message
0475	Identifier removed
0476	Machine is not on the network
0477	Object is remote
0478	The link has been severed
0479	Advertise error
047A	srmount error

Table 17. z/OS UNIX System Services sockets return codes (continued)	
Return code (Hexadecimal)	**Explanation**
047B	Communication error on send
047C	Protocol error
047D	Protocol error
047E	Cross mount point
047F	Remote address change
0480	The asynchronous I/O request has been canceled
0481	Socket send/receive gotten out of order
0482	Unattached streams error
0483	Streams push object error
0484	Streams closed error
0485	Streams link error
0486	Tcp error
Other	See the *z/OS UNIX System Services Messages and Codes* manual

APPC/MVS return codes

The tables in this section document the following return codes:

- APPC return codes
- APPC allocate services return codes
- APPC reason codes

See the *z/OS MVS Programming: Writing Transaction Programs for APPC/MVS* and *z/OS MVS Programming: Writing Servers for APPC/MVS* documentation for more information.

APPC return codes

This table documents the return codes that can be returned from APPC/MVS in messages from the distributed queuing component if you are using APPC/MVS as your communications protocol. These return codes can be returned to the local program in response to a call.

Table 18. APPC return codes and their meanings	
Return code (Hexadecimal)	**Explanation**
00	The call issued by the local program ran successfully. If the call specified a Notify_type of ECB, the call processing is performed asynchronously, and the ECB is posted when the processing is complete.
01	The caller specified an allocate_type that was other than *immediate*. Either APPC/MVS can not establish a session with the partner LU, or VTAM can not establish the conversation. In this case (when allocate_type is *immediate*), APPC/MVS converts this return code to "unsuccessful".
02	The conversation cannot be allocated on a session because of a condition that might be temporary. The program can try again the allocation request. The system returns this code when the allocate_type specified on a CMALLOC verb is other then *immediate*.

Table 18. APPC return codes and their meanings (continued)

Return code (Hexadecimal)	Explanation
03	The partner LU rejected the allocation request because the local program issued an Allocate call with the Conversation_type parameter set to either Basic_conversation or Mapped_conversation, and the partner program does not support the mapped or basic conversation protocol boundary. This return code is returned on a call made after the Allocate.
05	The partner LU rejected an ATBALLC or ATBALC2 (allocate) request because the partner program has one or more initialization parameter (PIP) variables defined. APPC/MVS does not support these parameters. This return code is returned on a call made after the Allocate. It is not returned for allocate requests made using CPI Communications.
06	The partner LU rejected the allocation request because the access security information is not valid. This return code is returned on a call subsequent to the Allocate.
08	The partner LU rejected the allocation request because the local program specified a synchronization level (with the Sync_level parameter) that the partner program does not support. This return code is returned on a call subsequent to the Allocate.
09	The partner LU rejected the allocation request because the local program specified a partner program that the partner LU does not recognize. This return code is returned on a call subsequent to the Allocate.
0A	The partner LU rejected the allocation request because the local program specified a partner program that the partner LU recognizes but cannot start. The condition is not temporary, and the program should not try again the allocation request. This return code is returned on a call subsequent to the Allocate.
0B	The partner LU rejected the allocation request because the local program specified a partner program that the partner LU recognizes but currently cannot start. The condition might be temporary, and the program can try again the allocation request. This return code is returned on a call subsequent to the Allocate.
11	The partner program issued a Deallocate call with a Deallocate_type of Deallocate_abend, or the partner LU has done so because of a partner program abnormal ending condition. If the partner program was in receive state when the call was issued, information sent by the local program and not yet received by the partner program is purged. This return code is reported to the local program on a call the program issues in Send or Receive state.
12	The partner program issued a Deallocate call on a basic or mapped conversation with a Deallocate_type of Deallocate_sync_level or Deallocate_flush. This return code is reported to the local program on a call the program issues in Receive state.

Table 18. APPC return codes and their meanings (continued)

Return code (Hexadecimal)	Explanation
13	The local program issued a call specifying an argument that was not valid. Specific reasons for the return code apply to the following callable services: **ATBALC2 or ATBALLC (LU 6.2 Allocate)** • The TP name was not 1 - 64 characters long • Either the SYMDEST name or the TP name length were not specified • SNASVCMG is specified as mode name • X'06' is used as the first character of a TP name • An SNA service TP name is used with a mapped conversation verb • The partner LU name was not valid • The mode name was not valid • The local LU name specified is either undefined or not permitted **CMALLC (CPI-C Allocate)** • SNASVCMG is specified as mode name • X'06'is used as the first character of a TP name • An SNA service TP name is used with a mapped conversation verb • The mode name was not valid
14	A product-specific error has been detected. The system writes symptom records that describe the error to SYS1.LOGREC.
15	Indicates one of the following: • The partner program made a Send_error call on a mapped conversation and the conversation for the partner program was in Send state. No truncation occurs at the mapped conversation protocol boundary. This return code is reported to the local program on a Receive call before receiving any data records or after receiving one or more data records. • The partner program made a Send_error call specifying the Type parameter with a value of PROG, the conversation for the partner program was in Send state, and the call did not truncate a logical record. No truncation occurs at the basic conversation protocol boundary when a program performs a Send_error before sending any logical records, or after sending a complete logical record. This return code is reported to the local program on a Receive call before receiving any logical records or after receiving one or more complete logical records.
16	The partner program made a Send_error call on a mapped conversation, or made a Send_error call on a basic conversation specifying the Type parameter with a value of PROG, and the conversation for the partner program was in Receive or Confirm state. The call might have caused information to be purged. Purging occurs when a program issued Send_error in receive state before receiving all the information sent by its partner program. No purging occurs when a program issues the call in Confirm state or in Receive state after receiving all the information sent by its partner program. The return code is normally reported to the local program on a call it issues before sending any information, depending on the call and when it is made.

Table 18. APPC return codes and their meanings (continued)

Return code (Hexadecimal)	Explanation
17	The partner program made a Send_error call specifying the Type parameter with a value of PROG, the conversation for the partner program was in Send state, and the call truncated a logical record. Truncation occurs at the basic conversation protocol boundary when a program begins sending a logical record and then makes a Send_error call before sending the complete logical record. This return code is reported to the local program on a Receive call it issues after receiving the truncated logical record.
18	The local program issued a call in which a programming error has been found in one or more parameters. Specific reasons for the return code apply to the following callable services: **ATBALC2 or ATBALLC (LU 6.2 Allocate)** • An unauthorized caller passed a nonzero TP_ID • For Sec_pgm-type security, both the user ID and password were not specified • For Sec_Pgm-type security, a user ID was specified with a blank password, or a password was specified with a blank user ID • The SYMDEST name was not found in the side information • The specified TP_ID is not associated with the address space • An unauthorized caller specified a Notify_Type of ECB **ATBCFM (LU 6.2 Allocate)** • An unauthorized caller specified a Notify_type of ECB • The Sync_Level field for the conversation was equal to sync_level_none **ATBDEAL (LU 6.2 Allocate)** • A Deallocate_type of deallocate_confirm was specified, and the Sync_Level field for the conversation was equal to sync_level_none **ATBPTR (LU 6.2 Prepare to Receive)** • A Prepare_To_Receive_Type of Prep_to_receive_sync_level was specified, and the Sync_Level field for the conversation was equal to sync_level_none **ATBSEND (LU 6.2 Send)** • The value in the 2 byte LL field was not valid • A Send_Type of Send_and_Confirm was specified, and the Sync_Level field for the conversation was equal to sync_level_none **CMINIT (CPI-C Initialize Conversation)** The SYMDEST name was not found in the side information

Table 18. APPC return codes and their meanings (continued)

Return code (Hexadecimal)	Explanation
19	The local program issued a call in a state that was not valid for that call. The program should not examine any other returned variables associated with the call as nothing is placed in the variables. The state of the conversation remains unchanged. If the error occurs in one of the following callable services, the conversation was in send state and the program started, but the program did not finish sending a logical record: • ATBCFM (LU 6.2 Allocate) • ATBDEAL (LU 6.2 Allocate) • ATBPTR (LU 6.2 Allocate) • ATBRCVW and ATBRCVI (LU 6.2 Receive and Wait and Receive Immediate) • ATBSEND (LU 6.2 Send)
1A	A failure occurred that caused the conversation to be prematurely terminated. The condition is not temporary, and the program should not try the transaction again until the condition is corrected.
1B	A failure occurred that caused the conversation to be prematurely terminated. The condition might be temporary, and the program can try the transaction again.
1C	The call issued by the local program did not run successfully. This return code is returned on the unsuccessful call. If this code is returned by the ATBRCVI (LU 6.2 Receive_Immediate) callable service, there is no data to be returned.
1E	The partner program issued a Deallocate call with a Deallocate_type of Deallocate_abend_SVC. If the partner program was in Receive state when the call was issued, information sent by the local program and not yet received by the partner program is purged. This return code is reported to the local program on a call the program issues in Send or Receive state.
1F	The partner program issued a Deallocate call with a Deallocate_type of Deallocate_abend_timer. If the partner program was in Receive state when the call was issued, information sent by the local program and not yet received by the partner program is purged. This return code is reported to the local program on a call the program issues in Send or Receive state.
20	The partner program issued a Send_error call specifying a Type parameter of SVC, the conversation for the partner program was in Send state, and the call did not truncate a logical record. This return code is returned on a Receive call. It is not returned for Send_error requests using CPI Communications.

Table 18. APPC return codes and their meanings (continued)	
Return code (Hexadecimal)	**Explanation**
21	The partner program issued a Send_error call specifying a Type parameter of SVC, the conversation for the partner program was in Receive, Confirm, or Sync_Point state, and the call might have caused information to be purged. This return code is normally returned to the local program on a call that the local program issues after sending some information to the partner program. However the return code can be returned on a call that the local program issues before sending any information, depending on when the call is issued. This code is not returned for Send_error requests using CPI Communications.
22	The partner program issued a Send_error call specifying a Type parameter of SVC, the conversation for the partner program was in Send state, and the call truncated a logical record. Truncation occurs when a program begins sending a logical record and then issues Send_error before sending the complete record. This return code is returned to the local program on a Receive call that the local program issues after receiving the truncated logical record. The code is not returned for Send_error requests using CPI Communications.
40	APPC/MVS is not currently active. Call the service again after APPC is available.
Other	See the *z/OS MVS Programming: Writing Transaction Programs for APPC/MVS* and *z/OS MVS Programming: Writing Servers for APPC/MVS* documentation for more information.

APPC allocate services return codes

This table documents the return codes that can be returned from APPC/MVS allocate queue services in messages from the distributed queuing component if you are using APPC/MVS as your communications protocol.

Table 19. APPC allocate services return codes and their meanings	
Return code (Hex)	**Explanation**
0	The service completed as requested.
4	The service completed, but possibly not as expected. See the reason code parameter for a description of the warning condition.
8	A user-supplied parameter was found to be in error. For example, a parameter contains characters not in the required character set. See the reason code parameter to determine which parameter is in error.
10	The service was unsuccessful. The cause is most likely a parameter error other than a syntax error, or an environmental error. For example, a syntactically valid LU name was specified, but the LU is not defined to APPC/MVS. An example of an environmental error is that the caller called the service while holding locks. See the reason code parameter for the specific cause of the error, and to determine whether the error can be corrected and the service issued again.

Table 19. APPC allocate services return codes and their meanings (continued)	
Return code (Hex)	**Explanation**
20	APPC/MVS service failure. Record the return and reason code, and give them to your system programmer, who should contact IBM Support.
40	APPC/MVS is not currently active. Call the service again after APPC is available.
Other	See the *z/OS MVS Programming: Writing Transaction Programs for APPC/MVS* and *z/OS MVS Programming: Writing Servers for APPC/MVS* documentation for more information.

APPC reason codes

This table documents the reason codes that can be returned from APPC/MVS allocate queue services in messages from the distributed queuing component if you are using APPC/MVS as your communications protocol.

Note: Some of the APPC return codes are not accompanied by a reason code; in these cases, the value in the reason code field can be ignored. See the documentation shown in "APPC/MVS return codes" on page 1136 for more information.

Table 20. APPC reason codes and their meanings	
Return code (Hex)	**Explanation**
1	The address space issued a Register_For_Allocates call that duplicated a previous Register_For_Allocate call (that is, the values specified for TP name, local LU name, partner LU name, user ID, and profile all matched those specified on a previous call to the Register_For_Allocates service).
2	A TP name is required, but none was specified.
3	The specified TP name contains characters that are not valid
4	The specified TP name length is outside the allowable range.
5	A local LU name is required, but none was specified.
7	An asynchronous call failed because a specified parameter was found to be inaccessible.
8	The caller held one or more locks when calling the service.
0A	A transaction scheduler called the Register_For_Allocate service, which is not allowed
0B	The specified symbolic destination name can not be found in the side information data set.
0C	The specified local LU is undefined.
0D	The specified local LU is not receiving inbound allocate requests.
0E	The Register_For_Allocate service was called, but the caller is not authorized to serve the specified TP name on the specified local LU.
0F	The specified local LU is inaccessible to the caller.
10	The service failed because of an APPC failure.
11	The specified allocate queue token does not represent an allocate queue for which this address space is registered.
12	The specified notify type is not valid.

Table 20. APPC reason codes and their meanings (continued)	
Return code (Hex)	**Explanation**
13	The specified timeout value is not valid.
14	The request was canceled while in progress. This might have been caused by a call to the Unregister_For_Allocates service, or the termination of the caller's address space.
15	A Receive_Allocate call completed, but no allocate request was available to be received.
1A	The specified event notification type is not valid.
1B	The specified event code is not supported or is not valid for this service.
1C	The netid retrieved from the side information data set does not match the local netid.
1D	The specified event code qualifier is not valid or supported.
1E	The Get_Event call completed, but no event element was available to be received.
1F	The call to the Get_Event service was interrupted because all event notification requests were canceled for this address space.
20	The call to the Get_Event service was rejected because a previous Get_Event call is currently outstanding.
21	The Get_Event call was rejected because no event notification is in effect for this address space.
22	The specified allocate queue keep time is outside the allowable range.
24	A call to the Unregister_For_Allocates service specified "unregister all" (that is, the allocate_queue_token was set to binary zeros), but this address space is not registered for any allocate queues.
25	The specified event get type is not valid.
26	The specified receive allocate type is not valid.
27	APPC/MVS cannot determine if the specified netid is valid.
29	The service failed because the supplied buffer was not large enough to contain the requested information.
Other	See the *z/OS MVS Programming: Writing Transaction Programs for APPC/MVS* and *z/OS MVS Programming: Writing Servers for APPC/MVS* documentation for more information.

z/OS Transport Layer Security (TLS) return codes for z/OS

IBM MQ for z/OS can use TLS with the various communication protocols. Use this topic to identify the error codes that can be returned by TLS.

Table 21 on page 1144 in this appendix documents the return codes, in decimal form, from the TLS that can be returned in messages from the distributed queuing component.

Table 22 on page 1146 in this appendix documents the return codes, in hexadecimal form, from the TLS function 'gsk_fips_state_set' that can be returned in messages from the distributed queuing component.

If the return code is not listed, or if you want more information, see SSL Function Return Codes in *z/OS Cryptographic Services System SSL Programming*.

Table 21. SSL return codes	
Return code (decimal)	**Explanation**
1	Handle is not valid.
3	An internal error has occurred.
4	Insufficient storage is available
5	Handle is in the incorrect state.
6	Key label is not found.
7	No certificates available.
8	Certificate validation error.
9	Cryptographic processing error.
10	ASN processing error.
11	LDAP processing error.
12	An unexpected error has occurred.
102	Error detected while reading key database or SAF key ring.
103	Incorrect key database record format.
106	Incorrect key database password.
109	No certificate authority certificates.
201	No key database password supplied.
202	Error detected while opening the key database.
203	Unable to generate temporary key pair
204	Key database password is expired.
302	Connection is active.
401	Certificate is expired or is not valid yet.
402	No TLS cipher specifications.
403	No certificate received from partner.
405	Certificate format is not supported.
406	Error while reading or writing data.
407	Key label does not exist.
408	Key database password is not correct.
410	TLS message format is incorrect.
411	Message authentication code is incorrect.
412	TLS protocol or certificate type is not supported.
413	Certificate signature is incorrect.
414	Certificate is not valid.
415	TLS protocol violation.
416	Permission denied.

Table 21. SSL return codes (continued)	
Return code (decimal)	**Explanation**
417	Self-signed certificate cannot be validated.
420	Socket closed by remote partner.
421	SSL 2.0 cipher is not valid.
422	SSL 3.0 cipher is not valid.
427	LDAP is not available.
428	Key entry does not contain a private key.
429	SSL 2.0 header is not valid.
431	Certificate is revoked.
432	Session renegotiation is not allowed.
433	Key exceeds allowable export size.
434	Certificate key is not compatible with cipher suite.
435	certificate authority is unknown.
436	Certificate revocation list cannot be found.
437	Connection closed.
438	Internal error reported by remote partner.
439	Unknown alert received from remote partner.
440	Incorrect key usage.
442	Multiple certificates exist for label.
443	Multiple keys are marked as the default.
444	Error encountered generaing random bytes.
445	Key database is not a FIPS mode database.
446	TLS extension mismatch has been encountered.
447	Required TLS extension has been rejected.
448	Requested server name is not recognized.
449	Unsupported fragment length was received.
450	TLS extension length field is not valid.
451	Elliptic Curve is not supported.
452	EC Parameters not supplied.
453	Signature not supplied.
454	Elliptic Curve parameters are not valid.
455	ICSF services are not available.
456	ICSF callable services returned a error.
457	ICSF PKCS#11 not operating in FIPS mode.
458	The SSL 3.0 expanded cipher is not valid.

Table 21. SSL return codes (continued)	
Return code (decimal)	**Explanation**
459	Elliptic Curve is not supported in FIPS mode.
460	Required TLS Renegotiation Indication not received.
461	EC domain parameter format is not supported.
462	Elliptic Curve point format is not supported.
463	Cryptographic hardware does not support service or algorithmn.
464	Elliptic curve list is not valid.
466	Signature algorithm pairs list is not valid.
467	Signature algorithm not in signature algorithm pairs list.
468	Certificate key algorithm not in signature algorithm pairs list.
501	Buffer size is not valid.
502	Socket request would block.
503	Socket read request would block.
504	Socket write request would block.
505	Record overflow.
601	Protocol is not TLS 1.0, TLS 1.1, or TLS 1.2.
602	Function identifier is not valid.
603	Specified function enumeration is not valid.
604	Send sequence number is near maxumum value.
701	Attribute identifier is not valid.
702	Attribute length is not valid.
703	Enumeration is not valid.
704	Session identifier cache callback is not valid.
705	Numeric value is not valid.
706	Attribute parameter is not valid.
707	TLS extension type is not valid.
708	Supplied TLS extension data is not valid.

Table 22. SSL return codes from 'gsk_fips_state_set'	
Return code (hexadecimal)	**Explanation**
03353050	The enumeration value is not valid or it cannot be set due to the current state.
0335306B	The System SSL FIPS mode state cannot be changed to FIPS mode because it is currently not in FIPS mode.

Table 22. SSL return codes from 'gsk_fips_state_set' (continued)	
Return code (hexadecimal)	**Explanation**
0335306C	The request to execute in FIPS mode failed because the Cryptographic Services Security Level 3 FMID is not installed so that the required System SSL DLLs could not be loaded.
03353067	The power on known answer tests failed. FIPS mode cannot be set.

z/OS Distributed queuing message codes

Distributed queuing is one of the components of IBM MQ for z/OS. Use this topic to interpret the message codes issued by the distributed queuing component.

Distributed queuing message codes are in the form *s* 0009 *nnn* (in hexadecimal). The error they identify is described in detail by error message CSQX *nnn*, although there are some exceptions. The following table shows the full correspondence. Distributed queuing message codes are used in some error messages, and in the event data for the MQRC_CHANNEL_STOPPED event. The event data also contains message inserts. The meanings of the inserts depend on the message code, and are shown in the following table, in the form in which they are given in the message explanation. Where no meaning is shown, the insert is not relevant to the message code, and the value set in the event message is unpredictable.

Note: *trptype* can be shown in various forms:

Message insert
Event data

TCP
TCP/IP

LU62
LU 6.2, APPC, CPI-C

Table 23. Distributed queuing message codes, and their corresponding message numbers, integer inserts, and character inserts.

Message code (*nnn*)	Message number	Integer insert 1	Integer insert 2	Character insert 1	Character insert 2	Character insert 3
001	CSQX501I			channel-name		
181	CSQX181E	response		exit-name		
182	CSQX182E	response		exit-name		
184	CSQX184E	address		exit-name		
189	CSQX189E	length		exit-name		
196	CSQX196E	data-length	agent-buffer length	exit-name		
197	CSQX197E	data-length	exit-buffer length	exit-name		
201	CSQX201E	return-code		conn-id	trptype	
202	CSQX202E	return-code		conn-id	trptype	
203	CSQX203E	return-code		conn-id	trptype	
204	CSQX204E	return-code		conn-id	trptype	
205	CSQX205E	return-code		conn-id	trptype	

Table 23. Distributed queuing message codes, and their corresponding message numbers, integer inserts, and character inserts. (continued)

Message code (*nnn*)	Message number	Integer insert 1	Integer insert 2	Character insert 1	Character insert 2	Character insert 3
206	CSQX206E	return-code		conn-id	trptype	
207	CSQX207E			conn-id	trptype	
208	CSQX208E	return-code		conn-id	trptype	
209	CSQX209E			conn-id	trptype	
211	CSQX027E					
212	CSQX212E	return-code				
213	CSQX213E	return-code			trptype	
237	CSQX203E	return-code	reason	conn-id	trptype	
238	CSQX213E	return-code	reason		trptype	
403	CSQX403I			channel-name	exit-name	
496	CSQX496I			channel-name		
498	CSQX498E	fieldvalue		channel-name		
506	CSQX506E			channel-name		
510	CSQX037E	mqrc			name	
511	CSQX038E	mqrc			name	
514	CSQX514E			channel-name		
519	CSQX519E			channel-name		
520	CSQX520E			channel-name		
525	CSQX525E			channel-name		
526	CSQX526E	msg-seqno	exp-seqno	channel-name		
527	CSQX527E			channel-name		
528	CSQX528I			channel-name		
533	CSQX533I			channel-name		
534	CSQX534E			channel-name		
536	CSQX536I			channel-name	exit-name	
540	CSQX540E	mqrc		commit identifier which includes channel-name		
542	the queue manager is stopping (no corresponding error message)					

Table 23. Distributed queuing message codes, and their corresponding message numbers, integer inserts, and character inserts. (continued)

Message code (*nnn*)	Message number	Integer insert 1	Integer insert 2	Character insert 1	Character insert 2	Character insert 3
544	see integer insert 1	1 - see message CSQX548E 2 - see message CSQX544E		channel-name		
545	CSQX545I			channel-name		
546	code 00E70546					
558	CSQX558E			channel-name		
565	CSQX565E			channel-name	qmgr-name	
569	CSQX569E			channel-name		
570	CSQX570E			channel-name		
572	CSQX572E			channel-name		
573	CSQX573E			channel-name		
574	CSQX574I			channel-name		
575	CSQX575E					
613	CSQX613E			channel-name		
620	CSQX620E	return-code		SSL-function		
631	CSQX631E			channel-name	local cipher spec	remote cipher spec
633	CSQX633E			channel-name		
634	CSQX634E			channel-name		
635	CSQX635E			channel-name		cipher spec
636	CSQX636E			channel-name	dist-name	
637	CSQX637E			channel-name		
638	CSQX638E			channel-name		
639	CSQX639E			channel-name		
640	CSQX640E			channel-name		key-name
641	CSQX641E			channel-name		
642	CSQX642E			channel-name		
643	CSQX643E			channel-name		
644	CSQX644E			channel-name		
999	CSQX599E			channel-name		

z/OS Queued Publish/Subscribe message codes

Queued Publish/Subscribe is a component of IBM MQ for z/OS. Use this topic to interpret the message codes issued by the queued Publish/Subscribe component.

Queued publish/subscribe message codes are in the form 5 *nnn* (in hexadecimal), and the error they identify is described in detail by error message CSQT *nnn*, although there are some exceptions. The following table shows the full correspondence. Queued publish/subscribe message codes are used in some error messages.

Table 24. Queued publish/subscribe message codes and any equivalent error messages

Message Code (*nnn*)	**Message Number**	**Description**
800	No equivalent message	Unexpected error
87F	CSQX036E	Failed

z/OS Messages from other products

Software products on the z/OS platform issue messages and each product uses a unique identifier. Use this topic to identify the different z/OS products using the unique identifier.

The following table shows the message prefixes for other products that you might receive while using IBM MQ for z/OS.

Table 25. Message prefixes

Prefix	**Component**	**Procedure**
AMQ	IBM MQ (not z/OS)	Consult Messages and reason codes
ATB	APPC	Consult *MVS System Messages*
ATR	Resource recovery services	Consult *MVS System Messages*
CBC	C/C++	Consult *C/MVS User's Guide*
CEE	Language Environment	Consult *Language Environment for z/OS Debugging Guide and Runtime Messages*
CSQ	IBM MQ for z/OS	Consult this documentation
CSV	Contents supervision	Consult *MVS System Messages*
DFH	CICS	Consult *CICS Messages and Codes*
DFS	IMS	Consult *IMS Messages and Codes*
DSN	Db2	Consult *Db2 Messages and Codes*
EDC	Language Environment	Consult *Language Environment for z/OS Debugging Guide and Runtime Messages*
EZA, EZB, EZY	TCP/IP	Consult *TCP/IP for MVS Messages and Codes*
IBM	Language Environment	Consult *Language Environment for z/OS Debugging Guide and Runtime Messages*
ICH	RACF	Consult *RACF Messages and Codes*
IDC	Access method services	Consult *MVS System Messages*
IEA	z/OS system services	Consult *MVS System Messages*
IEC	Data management services	Consult *MVS System Messages*
IEE,IEF	z/OS system services	Consult *MVS System Messages*

Table 25. Message prefixes (continued)		
Prefix	**Component**	**Procedure**
IKJ	TSO	Consult *MVS System Messages*
IST	VTAM	Consult *VTAM Messages and Codes*
IWM	z/OS workload management services	Consult *MVS System Messages*
IXC	Cross-system coupling facility (XCF)	Consult *MVS System Messages*
IXL	Cross-system extended services (XES)	Consult *MVS System Messages*

See the Message directory for a full list of the z/OS message code prefixes.

API completion and reason codes

For each call, a completion code and a reason code are returned by the queue manager or by an exit routine, to indicate the success or failure of the call.

For more information about the IBM MQ API, see Developing applications, and the reference information in Developing applications reference.

API completion codes

The following is a list of the completion codes (MQCC) returned by IBM MQ

0: Successful completion (MQCC_OK)

The call completed fully; all output parameters have been set.

The **Reason** parameter always has the value MQRC_NONE in this case.

1: Warning (partial completion) (MQCC_WARNING)

The call completed partially. Some output parameters might have been set in addition to the *CompCode* and *Reason* output parameters.

The **Reason** parameter gives additional information.

2: Call failed (MQCC_FAILED)

The processing of the call did not complete, and the state of the queue manager is normally unchanged; exceptions are specifically noted. Only the *CompCode* and *Reason* output parameters have been set; all other parameters are unchanged.

The reason might be a fault in the application program, or it might be a result of some situation external to the program, for example the application's authority might have been revoked. The **Reason** parameter gives additional information.

API reason codes

The reason code parameter (**Reason**) is a qualification to the completion code parameter (**CompCode**).

If there is no special reason to report, MQRC_NONE is returned. A successful call returns MQCC_OK and MQRC_NONE.

If the completion code is either MQCC_WARNING or MQCC_FAILED, the queue manager always reports a qualifying reason; details are given under each call description.

Where user exit routines set completion codes and reasons, they should adhere to these rules. In addition, any special reason values defined by user exits should be less than zero, to ensure that they do

not conflict with values defined by the queue manager. Exits can set reasons already defined by the queue manager, where these are appropriate.

Reason codes also occur in:

- The *Reason* field of the MQDLH structure
- The *Feedback* field of the MQMD structure

Where multiple client connection attempts are made, for example with a channel with a comma separated connection name and the connection sequence fails, reason codes other than MQRC_HOST_NOT_AVAILABLE and MQRC_STANDBY_Q_MGR are given priority when being returned to an application. In previous versions of the product, the last reason code encountered in the connection sequence is always returned, this could lead to configuration errors such as MQRC_NOT_AUTHORIZED encountered earlier in the connection attempt to not be returned to the application.

Each reason code in its entirety is supported on all the platforms that IBM MQ supports, unless there:

- Are one or more platform icons at the top of a reason code, in which case that reason code applies only to those platforms shown.
- Is a platform icon displayed against a specific statement, in which case that statement applies only to that platform.

See platform icons for a description of the icons.

The child topics are a set of reason codes, in numeric order, providing detailed information to help you understand them, including:

- an explanation of the circumstances that have caused the code to be raised
- the associated completion code
- suggested programmer actions in response to the code

0 (0000) (RC0): MQRC_NONE

Explanation

The call completed normally. The completion code (*CompCode*) is MQCC_OK.

Completion code

MQCC_OK

Programmer response

None.

900 (0384) (RC900): MQRC_APPL_FIRST

Explanation

This is the lowest value for an application-defined reason code returned by a data-conversion exit. Data-conversion exits can return reason codes in the range MQRC_APPL_FIRST through MQRC_APPL_LAST to indicate particular conditions that the exit has detected.

Completion code

MQCC_WARNING or MQCC_FAILED

Programmer response

As defined by the writer of the data-conversion exit.

999 (03E7) (RC999): MQRC_APPL_LAST

Explanation

This is the highest value for an application-defined reason code returned by a data-conversion exit. Data-conversion exits can return reason codes in the range MQRC_APPL_FIRST through MQRC_APPL_LAST to indicate particular conditions that the exit has detected.

Completion code

MQCC_WARNING or MQCC_FAILED

Programmer response

As defined by the writer of the data-conversion exit.

2001 (07D1) (RC2001): MQRC_ALIAS_BASE_Q_TYPE_ERROR

Explanation

An MQOPEN or MQPUT1 call was issued specifying an alias queue as the destination, but the **BaseQName** in the alias queue definition resolves to a queue that is not a local queue, a local definition of a remote queue, or a cluster queue.

A queue in a distribution list contains an alias queue that points to a topic object.

This reason code can occur when running the REFRESH CLUSTER command. See Application issues seen when running REFRESH CLUSTER

Completion code

MQCC_FAILED

Programmer response

Correct the queue definitions.

This reason code is also used to identify the corresponding event message “Alias Base Queue Type Error” on page 128.

2002 (07D2) (RC2002): MQRC_ALREADY_CONNECTED

Explanation

An MQCONN or MQCONNX call was issued, but the application is already connected to the queue manager.

- z/OS On z/OS, this reason code occurs for batch and IMS applications only; it does not occur for CICS applications.
- Multi On Multiplatforms, this reason code occurs if the application attempts to create a nonshared handle when a nonshared handle exists for the thread. A thread can have no more than one nonshared handle.

- Multi On Multiplatforms, this reason code occurs if an MQCONN call is issued from within an MQ channel exit, API Crossing Exit, or Async Consume Callback function, and a shared hConn is bound to this thread.
- Multi On Multiplatforms, this reason code occurs if an MQCONNX call that does not specify one of the MQCNO_HANDLE_SHARE_* options is issued from within an MQ channel exit, API Crossing Exit, or Async Consume Callback function, and a shared hConn is bound to this thread
- Windows On Windows, MTS objects do not receive this reason code, as additional connections to the queue manager are permitted.

Completion code

MQCC_WARNING

Programmer response

None. The **Hconn** parameter returned has the same value as was returned for the previous MQCONN or MQCONNX call.

An MQCONN or MQCONNX call that returns this reason code does not mean that an additional MQDISC call must be issued to disconnect from the queue manager. If this reason code is returned because the application has been called in a situation where the MQCONN has already been done, do not issue a corresponding MQDISC, because this causes the application that issued the original MQCONN or MQCONNX call to be disconnected as well.

2003 (07D3) (RC2003): MQRC_BACKED_OUT

Explanation

The current unit of work encountered an unrecoverable error or was backed out. This reason code is issued in the following cases:

- On an MQCMIT or MQDISC call, when the commit operation fails and the unit of work is backed out. All resources that participated in the unit of work are returned to their state at the start of the unit of work. The MQCMIT or MQDISC call completes with MQCC_WARNING in this case.
  - z/OS On z/OS, this reason code occurs only for batch applications.
- On an MQGET, MQPUT, or MQPUT1 call that is operating within a unit of work, when the unit of work already encountered an error that prevents the unit of work from being committed (for example, when the log space is exhausted). The application must issue the appropriate call to back out the unit of work. (For a unit of work that is coordinated by the queue manager, this call is the MQBACK call, although the MQCMIT call has the same effect in these circumstances.) The MQGET, MQPUT, or MQPUT1 call completes with MQCC_FAILED in this case.
  - z/OS On z/OS, this case does not occur.
- On an asynchronous consumption callback (registered by an MQCB call), the unit of work is backed out and the asynchronous consumer should call MQBACK. The asynchronous consumer should also interrogate the State field of the MQCBC (call back context) provided, to see whether the connection has been suspended, and resume if required.
  - z/OS On z/OS, this case does not occur.

Completion code

MQCC_WARNING or MQCC_FAILED

Programmer response

Check the returns from previous calls to the queue manager. For example, a previous MQPUT call might have failed.

2004 (07D4) (RC2004): MQRC_BUFFER_ERROR

Explanation

The **Buffer** parameter is not valid for one of the following reasons:

- The parameter pointer is not valid. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)
- The parameter pointer points to storage that cannot be accessed for the entire length specified by **BufferLength**.
- For calls where **Buffer** is an output parameter: the parameter pointer points to read-only storage.

Completion code

MQCC_FAILED

Programmer response

Correct the parameter.

2005 (07D5) (RC2005): MQRC_BUFFER_LENGTH_ERROR

Explanation

The **BufferLength** parameter is not valid, or the parameter pointer is not valid. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)

This reason can also be returned to an MQ MQI client program on the MQCONN or MQCONNX call if the negotiated maximum message size for the channel is smaller than the fixed part of any call structure.

This reason should also be returned by the MQZ_ENUMERATE_AUTHORITY_DATA installable service component when the **AuthorityBuffer** parameter is too small to accommodate the data to be returned to the invoker of the service component.

This reason code can also be returned when a zero length multicast message has been supplied where a positive length is required.

Completion code

MQCC_FAILED

Programmer response

Specify a value that is zero or greater. For the mqAddString and mqSetString calls, the special value MQBL_NULL_TERMINATED is also valid.

2006 (07D6) (RC2006): MQRC_CHAR_ATTR_LENGTH_ERROR

Explanation

CharAttrLength is negative (for MQINQ or MQSET calls), or is not large enough to hold all selected attributes (MQSET calls only). This reason also occurs if the parameter pointer is not valid. (It is not

always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)

Completion code

MQCC_FAILED

Programmer response

Specify a value large enough to hold the concatenated strings for all selected attributes.

2007 (07D7) (RC2007): MQRC_CHAR_ATTRS_ERROR

Explanation

CharAttrs is not valid. The parameter pointer is not valid, or points to read-only storage for MQINQ calls or to storage that is not as long as implied by **CharAttrLength**. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)

Completion code

MQCC_FAILED

Programmer response

Correct the parameter.

2008 (07D8) (RC2008): MQRC_CHAR_ATTRS_TOO_SHORT

Explanation

For MQINQ calls, **CharAttrLength** is not large enough to contain all of the character attributes for which MQCA_* selectors are specified in the **Selectors** parameter.

The call still completes, with the **CharAttrs** parameter string filled in with as many character attributes as there is room for. Only complete attribute strings are returned: if there is insufficient space remaining to accommodate an attribute in its entirety, that attribute and subsequent character attributes are omitted. Any space at the end of the string not used to hold an attribute is unchanged.

An attribute that represents a set of values (for example, the namelist **Names** attribute) is treated as a single entity—either all of its values are returned, or none.

Completion code

MQCC_WARNING

Programmer response

Specify a large enough value, unless only a subset of the values is needed.

2009 (07D9) (RC2009): MQRC_CONNECTION_BROKEN

Explanation

Connection to the queue manager has been lost. This can occur because the queue manager has ended. If the call is an MQGET call with the MQGMO_WAIT option, the wait has been canceled. All connection and object handles are now invalid.

For MQ MQI client applications, it is possible that the call did complete successfully, even though this reason code is returned with a CompCode of MQCC_FAILED.

Completion code

MQCC_FAILED

Programmer response

Applications can attempt to reconnect to the queue manager by issuing the MQCONN or MQCONNX call. It might be necessary to poll until a successful response is received.

- z/OS On z/OS for CICS applications, it is not necessary to issue the MQCONN or MQCONNX call, because CICS applications are connected automatically.

Any uncommitted changes in a unit of work should be backed out. A unit of work that is coordinated by the queue manager is backed out automatically.

z/OS For z/OS IMS check that the subsystem is started using the IMS DIS SUBSYS command, and if necessary start it using the IMS STA SUBSYS command.

Related concepts
IBM MQ and IMS

2010 (07DA) (RC2010): MQRC_DATA_LENGTH_ERROR

Explanation

The **DataLength** parameter is not valid. Either the parameter pointer is not valid, or it points to read-only storage. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)

This reason can also be returned to an MQ MQI client program on the MQGET, MQPUT, or MQPUT1 call, if the **BufferLength** parameter exceeds the maximum message size that was negotiated for the client channel. This might be because MAXMSGL is not set correctly for the channel definition (see Maximum message length (MAXMSGL)), or, if you are using MQCONNX, and supplying an MQCD, then the **MaxMsgLength** of that data structure needs to be set with a higher value (see Using MQCONNX).

Completion code

MQCC_FAILED

Programmer response

Correct the parameter.

If the error occurs for an MQ MQI client program, also check that the maximum message size for the channel is big enough to accommodate the message being sent; if it is not big enough, increase the maximum message size for the channel.

2011 (07DB) (RC2011): MQRC_DYNAMIC_Q_NAME_ERROR

Explanation

On the MQOPEN call, a model queue is specified in the ObjectName field of the **ObjDesc** parameter, but the DynamicQName field is not valid, for one of the following reasons:

- DynamicQName is completely blank (or blank up to the first null character in the field).
- Characters are present that are not valid for a queue name.

- An asterisk is present beyond the 33rd position (and before any null character).
- An asterisk is present followed by characters that are not null and not blank.

This reason code can also sometimes occur when a server application opens the reply queue specified by the `ReplyToQ` and `ReplyToQMgr` fields in the MQMD of a message that the server has just received. In this case the reason code indicates that the application that sent the original message placed incorrect values into the `ReplyToQ` and `ReplyToQMgr` fields in the MQMD of the original message.

Completion code

MQCC_FAILED

Programmer response

Specify a valid name.

2012 (07DC) (RC2012): MQRC_ENVIRONMENT_ERROR

Explanation

The call is not valid for the current environment.

- z/OS On z/OS, when one of the following applies:
  - An MQCONN or MQCONNX call was issued, but the application has been linked with an adapter that is not supported in the environment in which the application is running. For example, this can arise when the application is linked with the IBM MQ classes for Java RRS adapter, but the application is running in a Db2 Stored Procedure address space. RRS is not supported in this environment. Stored Procedures wishing to use the MQ RRS adapter must run in a Db2 WLM-managed Stored Procedure address space.
  - An MQCMIT or MQBACK call was issued, but the application has been linked with the RRS batch adapter CSQBRSTB. This adapter does not support the MQCMIT and MQBACK calls.
  - An MQCMIT or MQBACK call was issued in the CICS or IMS environment.
  - The RRS subsystem is not operational on the z/OS system that ran the application.
  - An MQCTL call with MQOP_START or an MQCB call registering an Event Listener, was issued but the application is not allowed to create a POSIX thread.
  - An IBM MQ classes for Java application has instantiated an MQQueueManager object using the CLIENT transport in an environment where it is not supported.

    The z/OS environment only supports IBM MQ 9.0.4 (and later) classes for Java applications, that use the CLIENT transport to connect to an IBM MQ 9.0.4 (and later) queue manager running on z/OS, that has **ADVCAP(ENABLED)**.

    See DISPLAY QMGR for more information on **ADVCAP**.
  - An IBM MQ classes for Java, or an IBM MQ classes for JMS, application using the CLIENT transport has accessed a queue that is protected by an Advanced Message Security for z/OS policy. The z/OS environment does not support Advanced Message Security for z/OS when using the CLIENT transport.
  - An IBM MQ classes for Java, or an IBM MQ classes for JMS application using client transport supplied a password longer than 12 characters when using compatibility mode to authenticate with the queue manager.
- Multi On Multiplatforms, when one of the following applies:
  - The application is linked to unsupported libraries.

  - The application is linked to the wrong libraries (threaded or nonthreaded). IBM MQ classes for Java client applications that are configured to use automatic reconnection must run in a threaded environment.
  - An MQBEGIN, MQCMIT, or MQBACK call was issued, but an external unit-of-work manager is in use. For example, this reason code occurs on Windows when an MTS object is running as a DTC transaction. This reason code also occurs if the queue manager does not support units of work.
  - The MQBEGIN call was issued in an IBM MQ MQI client environment.
  - An MQXCLWLN call was issued, but the call did not originate from a cluster workload exit.
  - An MQCONNX call was issued specifying the option MQCNO_HANDLE_SHARE_NONE from within an MQ channel exit, an API Exit, or a Callback function. The reason code occurs only if a shared hConn is bound to the application thread.
  - An IBM MQ Object is unable to connect fastpath.
  - An IBM MQ classes for Java application has created an MQQueueManager object that uses the CLIENT transport, and then called MQQueueManager.begin(). This method can only be called on MQQueueManager objects that use the BINDINGS transport.
  - An MQ client application that has been configured to use automatic reconnection attempted to connect using a channel defined with SHARECNV(0).
  - An IBM MQ classes for Java, or an IBM MQ classes for JMS application using client transport supplied a password longer than 12 characters when using compatibility mode to authenticate with the queue manager.
  - An MQCTL call or an MQCB call was issued in a MQI client environment, using a channel defined with SHARECNV(0).
- Windows On Windows, when using the managed .NET client, an attempt was made to use one of the unsupported features:
  - Unmanaged channel exits
  - XA Transactions
  - Communications other than TCP/IP
  - Channel compression

Completion code

MQCC_FAILED

Programmer response

Take one of the following actions:

- On z/OS:
  - Link the application with the correct adapter.
  - Modify the application to use the SRRCMIT and SRRBACK calls in place of the MQCMIT and MQBACK calls. Alternatively, link the application with the RRS batch adapter CSQBRRSI. This adapter supports MQCMIT and MQBACK in addition to SRRCMIT and SRRBACK.
  - For a CICS or IMS application, issue the appropriate CICS or IMS call to commit or backout the unit of work.
  - Start the RRS subsystem on the z/OS system that is running the application.
  - If your application uses Language Environment (LE) ensure that it uses the DLL interface and it runs with POSIX(ON).
  - Ensure that your application is allowed to use z/OS UNIX System Services (z/OS UNIX).
  - Ensure that your Connection Factory definitions for local z/OS applications and WebSphere Application Server applications use Transport Type with bindings mode connections.

Ensure that any client mode connections are made to supported queue managers, and do not access any queue protected by an IBM MQ Advanced Message Security for z/OS policy.

- In an IBM MQ classes for Java, or IBM MQ classes for JMS application that is using client bindings, use MQCSP authentication mode to supply a password longer than 12 characters when authenticating with a queue manager. For more information about connection authentication with Java client applications, see Connection authentication with the Java client.
- In the other environments:
  - Link the application with the correct libraries (threaded or nonthreaded).
  - Remove from the application the call or feature that is not supported.
  - Change your application to run setuid, if you want to run fastpath.

2013 (07DD) (RC2013): MQRC_EXPIRY_ERROR

Explanation

On an MQPUT or MQPUT1 call, the value specified for the `Expiry` field in the message descriptor MQMD is not valid.

This reason code is also generated by JMS applications specifying a delivery delay value greater than either the:

- Message expiry time specified by the application, or
- Expiry time set by the **CUSTOM(CAPEXPRY)** attribute of the objects used in the resolution of the target queue or topic.
- Expiry time set by the **CAPEXPRY** attribute of the objects used in the resolution of the target queue or topic.

Completion code

MQCC_FAILED

Programmer response

Specify a value that is greater than zero, or the special value MQEI_UNLIMITED.

Ensure that the delivery delay specified by JMS applications is less than the:

- Message expiry time specified by the application, or
- Expiry time set by the **CUSTOM(CAPEXPRY)** attribute of the objects used in the resolution of the target queue or topic.
- Expiry time set by the **CAPEXPRY** attribute of the objects used in the resolution of the target queue or topic.

2014 (07DE) (RC2014): MQRC_FEEDBACK_ERROR

Explanation

On an MQPUT or MQPUT1 call, the value specified for the `Feedback` field in the message descriptor MQMD is not valid. The value is not MQFB_NONE, and is outside both the range defined for system feedback codes and the range defined for application feedback codes.

Completion code

MQCC_FAILED

Programmer response

Specify MQFB_NONE, or a value in the range MQFB_SYSTEM_FIRST through MQFB_SYSTEM_LAST, or MQFB_APPL_FIRST through MQFB_APPL_LAST.

2016 (07E0) (RC2016): MQRC_GET_INHIBITED

Explanation

MQGET calls are currently inhibited for the queue, or for the queue to which this queue resolves.

Completion code

MQCC_FAILED

Programmer response

If the system design allows get requests to be inhibited for short periods, retry the operation later.

This reason code is also used to identify the corresponding event message "Get Inhibited" on page 193.

System programmer action

Use `ALTER QLOCAL(...) GET(ENABLED)` to allow messages to be got.

2017 (07E1) (RC2017): MQRC_HANDLE_NOT_AVAILABLE

Explanation

An MQOPEN, MQPUT1 or MQSUB call was issued, but the maximum number of open handles allowed for the current task has already been reached. Be aware that when a distribution list is specified on the MQOPEN or MQPUT1 call, each queue in the distribution list uses one handle.

- z/OS On z/OS, "task' means a CICS task, a z/OS task, or an IMS-dependent region.

In addition, the MQSUB call allocates two handles when you do not provide an object handle on input.

Completion code

MQCC_FAILED

Programmer response

Check whether the application is issuing MQOPEN calls without corresponding MQCLOSE calls. If it is, modify the application to issue the MQCLOSE call for each open object as soon as that object is no longer needed.

Also check whether the application is specifying a distribution list containing a large number of queues that are consuming all of the available handles. If it is, increase the maximum number of handles that the task can use, or reduce the size of the distribution list. The maximum number of open handles that a task can use is given by the **MaxHandles** queue manager attribute.

2018 (07E2) (RC2018): MQRC_HCONN_ERROR

Explanation

The connection handle Hconn is not valid, for one of the following reasons:

- The parameter pointer is not valid, or (for the MQCONN or MQCONNX call) points to read-only storage. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)
- The value specified was not returned by a preceding MQCONN or MQCONNX call.
- The value specified has been made invalid by a preceding MQDISC call.
- The handle is a shared handle that has been made invalid by another thread issuing the MQDISC call.
- The handle is a shared handle that is being used on the MQBEGIN call (only nonshared handles are valid on MQBEGIN).
- The handle is a nonshared handle that is being used a thread that did not create the handle.
- The call was issued in the MTS environment in a situation where the handle is not valid (for example, passing the handle between processes or packages; note that passing the handle between library packages is supported).
- The conversion program is not defined as OPENAPI, when the MQXCNVC call is invoked by running a character conversion exit program with CICS TS 3.2 or higher. When the conversion process runs, the TCB is switched to the Quasi Reentrant (QR) TCB, making the connection incorrect.

Completion code

MQCC_FAILED

Programmer response

Ensure that a successful MQCONN or MQCONNX call is performed for the queue manager, and that an MQDISC call has not already been performed for it. Ensure that the handle is being used within its valid scope (see the description of MQCONN in MQCONN for more information about MQCONN).

- z/OS On z/OS, also check that the application has been linked with the correct stub; this is CSQCSTUB for CICS applications, CSQBSTUB for batch applications, and CSQQSTUB for IMS applications. Also, the stub used must not belong to a release of the queue manager that is more recent than the release on which the application will run.

Ensure the character conversion exit program run by your CICS TS 3.2 or higher application, which invokes the MQXCNVC call, is defined as OPENAPI. This definition prevents the 2018 MQRC_HCONN_ERROR error caused by from an incorrect connection, and allows the MQGET to complete.

2019 (07E3) (RC2019): MQRC_HOBJ_ERROR

Explanation

The object handle `Hobj` is not valid, for one of the following reasons:

- The parameter pointer is not valid, or (for the MQOPEN call) points to read-only storage. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)
- The value specified was not returned by a preceding MQOPEN call.
- The value specified has been made invalid by a preceding MQCLOSE call.
- The handle is a shared handle that has been made invalid by another thread issuing the MQCLOSE call.
- The handle is a nonshared handle that is being used by a thread that did not create the handle.
- The call is MQGET or MQPUT, but the object represented by the handle is not a queue.

Completion code

MQCC_FAILED

Programmer response

Ensure that a successful MQOPEN call is performed for this object, and that an MQCLOSE call has not already been performed for it. Ensure that the handle is being used within its valid scope (see the description of MQOPEN in MQOPEN for more information).

2020 (07E4) (RC2020): MQRC_INHIBIT_VALUE_ERROR

Explanation

On an MQSET call, the value specified for either the MQIA_INHIBIT_GET attribute or the MQIA_INHIBIT_PUT attribute is not valid.

Completion code

MQCC_FAILED

Programmer response

Specify a valid value for the **InhibitGet** or **InhibitPut** queue attribute.

2021 (07E5) (RC2021): MQRC_INT_ATTR_COUNT_ERROR

Explanation

On an MQINQ or MQSET call, the **IntAttrCount** parameter is negative (MQINQ or MQSET), or smaller than the number of integer attribute selectors (MQIA_*) specified in the **Selectors** parameter (MQSET only). This reason also occurs if the parameter pointer is not valid. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)

Completion code

MQCC_FAILED

Programmer response

Specify a value large enough for all selected integer attributes.

2022 (07E6) (RC2022): MQRC_INT_ATTR_COUNT_TOO_SMALL

Explanation

On an MQINQ call, the **IntAttrCount** parameter is smaller than the number of integer attribute selectors (MQIA_*) specified in the **Selectors** parameter.

The call completes with MQCC_WARNING, with the **IntAttrs** array filled in with as many integer attributes as there is room for.

Completion code

MQCC_WARNING

Programmer response

Specify a large enough value, unless only a subset of the values is needed.

2023 (07E7) (RC2023): MQRC_INT_ATTRS_ARRAY_ERROR

Explanation

On an MQINQ or MQSET call, the **IntAttrs** parameter is not valid. The parameter pointer is not valid (MQINQ and MQSET), or points to read-only storage or to storage that is not as long as indicated by the **IntAttrCount** parameter (MQINQ only). (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)

Completion code

MQCC_FAILED

Programmer response

Correct the parameter.

2024 (07E8) (RC2024): MQRC_SYNCPOINT_LIMIT_REACHED

Explanation

An MQGET, MQPUT, or MQPUT1 call failed because it would have caused the number of uncommitted messages in the current unit of work to exceed the limit defined for the queue manager (see the **MaxUncommittedMsgs** queue manager attribute). The number of uncommitted messages is the sum of the following since the start of the current unit of work:

- Messages put by the application with the MQPMO_SYNCPOINT option
- Messages retrieved by the application with the MQGMO_SYNCPOINT option
- Trigger messages and COA report messages generated by the queue manager for messages put with the MQPMO_SYNCPOINT option
- COD report messages generated by the queue manager for messages retrieved with the MQGMO_SYNCPOINT option

When publishing messages out of syncpoint to topics it is possible to receive this reason code; see Publications under syncpoint for more information.

Completion code

MQCC_FAILED

Programmer response

Check whether the application is looping. If it is not, consider reducing the complexity of the application. Alternatively, increase the queue manager limit for the maximum number of uncommitted messages within a unit of work.

- z/OS On z/OS, the limit for the maximum number of uncommitted messages can be changed by using the **ALTER QMGR** command.
- IBM i On IBM i , the limit for the maximum number of uncommitted messages can be changed by using the **CHGMQM** command.

2025 (07E9) (RC2025): MQRC_MAX_CONNS_LIMIT_REACHED

Explanation

The MQCONN or MQCONNX call was rejected because the maximum number of concurrent connections has been exceeded.

- z/OS On z/OS, the connection limits are 32767 for both TSO and Batch.
- Multi On Multiplatforms, this reason code can also occur on the MQOPEN call.
- When using Java applications, the connection manager might define a limit to the number of concurrent connections.

  Note: The application using IBM MQ might have delegated the management of connections to a framework or connection pool, for example, a Java EE application server, an application framework such as Spring, an IBM Container (for IBM Cloud), or a combination of these. For more information, see IBM MQ classess for JMS object pooling.

Completion code

MQCC_FAILED

Programmer response

Either increase the size of the appropriate parameter value, or reduce the number of concurrent connections.

Related concepts

Connection pooling in IBM MQ classes for Java

2026 (07EA) (RC2026): MQRC_MD_ERROR

Explanation

The MQMD structure is not valid, for one of the following reasons:

- The `StrucId` field is not MQMD_STRUC_ID.
- The `Version` field specifies a value that is not valid or not supported.
- The parameter pointer is not valid. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)
- The queue manager cannot copy the changed structure to application storage, even though the call is successful. This can occur, for example, if the pointer points to read-only storage.

Completion code

MQCC_FAILED

Programmer response

Ensure that input fields in the MQMD structure are set correctly.

2027 (07EB) (RC2027): MQRC_MISSING_REPLY_TO_Q

Explanation

On an MQPUT or MQPUT1 call, the ReplyToQ field in the message descriptor MQMD is blank, but one or both of the following is true:

- A reply was requested (that is, MQMT_REQUEST was specified in the MsgType field of the message descriptor).
- A report message was requested in the Report field of the message descriptor.

Completion code

MQCC_FAILED

Programmer response

Specify the name of the queue to which the reply message or report message is to be sent.

2029 (07ED) (RC2029): MQRC_MSG_TYPE_ERROR

Explanation

Either:

- On an MQPUT or MQPUT1 call, the value specified for the MsgType field in the message descriptor (MQMD) is not valid.
- A message processing program received a message that does not have the expected message type. For example, if the IBM MQ command server receives a message which is not a request message (MQMT_REQUEST) then it rejects the request with this reason code.

Completion code

MQCC_FAILED

Programmer response

Specify a valid value for the MsgType field. In the case where a request is rejected by a message processing program, refer to the documentation for that program for details of the message types that it supports.

2030 (07EE) (RC2030): MQRC_MSG_TOO_BIG_FOR_Q

Explanation

An MQPUT or MQPUT1 call was issued to put a message on a queue, but the message was too long for the queue and MQMF_SEGMENTATION_ALLOWED was not specified in the MsgFlags field in MQMD. If segmentation is not allowed, the length of the message cannot exceed the lesser of the queue **MaxMsgLength** attribute and queue manager **MaxMsgLength** attribute.

- z/OS On z/OS, the queue manager does not support the segmentation of messages; if MQMF_SEGMENTATION_ALLOWED is specified, it is accepted but ignored.

This reason code can also occur when MQMF_SEGMENTATION_ALLOWED is specified, but the nature of the data present in the message prevents the queue manager splitting it into segments that are small enough to place on the queue:

- For a user-defined format, the smallest segment that the queue manager can create is 16 bytes.
- For a built-in format, the smallest segment that the queue manager can create depends on the particular format, but is greater than 16 bytes in all cases other than MQFMT_STRING (for MQFMT_STRING the minimum segment size is 16 bytes).

MQRC_MSG_TOO_BIG_FOR_Q can also occur in the Feedback field in the message descriptor of a report message; in this case it indicates that the error was encountered by a message channel agent when it attempted to put the message on a remote queue.

Completion code

MQCC_FAILED

Programmer response

Check whether the **BufferLength** parameter is specified correctly; if it is, do one of the following:

- Increase the value of the queue's **MaxMsgLength** attribute; the queue manager's **MaxMsgLength** attribute may also need increasing.
- Break the message into several smaller messages.
- Specify MQMF_SEGMENTATION_ALLOWED in the MsgFlags field in MQMD; this will allow the queue manager to break the message into segments.

2031 (07EF) (RC2031): MQRC_MSG_TOO_BIG_FOR_Q_MGR

Explanation

An MQPUT or MQPUT1 call was issued to put a message on a queue, but the message was too long for the queue manager and MQMF_SEGMENTATION_ALLOWED was not specified in the MsgFlags field in MQMD. If segmentation is not allowed, the length of the message cannot exceed the lesser of the queue manager **MaxMsgLength** attribute and queue **MaxMsgLength** attribute.

This reason code can also occur when MQMF_SEGMENTATION_ALLOWED is specified, but the nature of the data present in the message prevents the queue manager splitting it into segments that are small enough for the queue manager limit:

- For a user-defined format, the smallest segment that the queue manager can create is 16 bytes.
- For a built-in format, the smallest segment that the queue manager can create depends on the particular format, but is greater than 16 bytes in all cases other than MQFMT_STRING (for MQFMT_STRING the minimum segment size is 16 bytes).

MQRC_MSG_TOO_BIG_FOR_Q_MGR can also occur in the Feedback field in the message descriptor of a report message; in this case it indicates that the error was encountered by a message channel agent when it attempted to put the message on a remote queue.

This reason also occurs if a channel, through which the message is to pass, has restricted the maximum message length to a value that is actually less than that supported by the queue manager, and the message length is greater than this value.

Completion code

MQCC_FAILED

Programmer response

Check whether the **BufferLength** parameter is specified correctly; if it is, do one of the following:

- Increase the value of the queue manager's **MaxMsgLength** attribute; the queue's **MaxMsgLength** attribute may also need increasing.

- Break the message into several smaller messages.
- Specify MQMF_SEGMENTATION_ALLOWED in the MsgFlags field in MQMD; this will allow the queue manager to break the message into segments.
- Check the channel definitions.

2033 (07F1) (RC2033): MQRC_NO_MSG_AVAILABLE

Explanation

An MQGET call was issued, but there is no message on the queue satisfying the selection criteria specified in MQMD (the MsgId and CorrelId fields), and in MQGMO (the Options and MatchOptions fields). Either the MQGMO_WAIT option was not specified, or the time interval specified by the WaitInterval field in MQGMO has expired. This reason is also returned for an MQGET call for browse, when the end of the queue has been reached.

This reason code can also be returned by the mqGetBag and mqExecute calls. mqGetBag is similar to MQGET. For the mqExecute call, the completion code can be either MQCC_WARNING or MQCC_FAILED:

- If the completion code is MQCC_WARNING, some response messages were received during the specified wait interval, but not all. The response bag contains system-generated nested bags for the messages that were received.
- If the completion code is MQCC_FAILED, no response messages were received during the specified wait interval.

Completion code

MQCC_WARNING or MQCC_FAILED

Programmer response

If this is an expected condition, no corrective action is required.

If this is an unexpected condition, check that:

- The message was put on the queue successfully.
- The unit of work (if any) used for the MQPUT or MQPUT1 call was committed successfully.
- The options controlling the selection criteria are specified correctly. All of the following can affect the eligibility of a message for return on the MQGET call:
  - MQGMO_LOGICAL_ORDER
  - MQGMO_ALL_MSGS_AVAILABLE
  - MQGMO_ALL_SEGMENTS_AVAILABLE
  - MQGMO_COMPLETE_MSG
  - MQMO_MATCH_MSG_ID
  - MQMO_MATCH_CORREL_ID
  - MQMO_MATCH_GROUP_ID
  - MQMO_MATCH_MSG_SEQ_NUMBER
  - MQMO_MATCH_OFFSET
  - Value of MsgId field in MQMD
  - Value of CorrelId field in MQMD

Consider waiting longer for the message.

2034 (07F2) (RC2034): MQRC_NO_MSG_UNDER_CURSOR

Explanation

An MQGET call was issued with either the MQGMO_MSG_UNDER_CURSOR or the MQGMO_BROWSE_MSG_UNDER_CURSOR option. However, the browse cursor is not positioned at a retrievable message. This is caused by one of the following:

- The cursor is positioned logically before the first message (as it is before the first MQGET call with a browse option has been successfully performed).
- The message the browse cursor was positioned on has been locked or removed from the queue (probably by some other application) since the browse operation was performed.
- The message the browse cursor was positioned on has expired.

Completion code

MQCC_FAILED

Programmer response

Check the application logic. This may be an expected reason if the application design allows multiple servers to compete for messages after browsing. Consider also using the MQGMO_LOCK option with the preceding browse MQGET call.

2035 (07F3) (RC2035): MQRC_NOT_AUTHORIZED

General explanation

Explanation

The user of the application or channel that produced the error, is not authorized to perform the operation attempted:

- On an MQCONN or MQCONNX call, the user is not authorized to connect to the queue manager. This can be for one of the following reasons:
  - For locally bound applications, the application user ID has not been granted authority to connect to the queue manager.
  - An invalid user ID or password was specified in the MQCSP structure on an MQCONNX call.
  - An invalid authentication token was specified in the MQCSP structure on an MQCONNX call.
  - The queue manager is configured to require applications to provide authentication credentials in an MQCSP structure when connecting, but the application did not provide any credentials.

  z/OS On z/OS, for CICS applications, MQRC_CONNECTION_NOT_AUTHORIZED is issued instead.
- On an MQCONNX call, the length of the user ID or password is greater than the maximum length permitted. The maximum length of the user ID depends on the platform. For more information, see User IDs.
- On an attempt to connect to the queue manager in an IBM MQ classes for Java, or an IBM MQ classes for JMS application that is using client transport. This can be for one of the following reasons:
  - The application is using compatibility mode to authenticate with the queue manager, and the length of the user ID is greater than the maximum length permitted of 12 characters.
  - The authentication mode that is used by the application has changed after upgrading the IBM MQ classes for Java or IBM MQ classes for JMS to IBM MQ 9.3.0 or later. For more information about the authentication mode used by Java clients, see Connection authentication with the Java client.

- On an MQOPEN or MQPUT1 call, the user is not authorized to open the object for the option(s) specified.
  - z/OS On z/OS, if the object being opened is a model queue, this reason also arises if the user is not authorized to create a dynamic queue with the required name.
- On an MQCLOSE call, the user is not authorized to delete the object, which is a permanent dynamic queue, and the **Hobj** parameter that is specified on the MQCLOSE call is not the handle returned by the MQOPEN call that created the queue.
- On a command, the user is not authorized to issue the command, or to access the object it specifies.
- On an MQSUB call, the user is not authorized to subscribe to the topic.
- On an MQSUB call, using non-managed destination queues, the user is not authorized to use the destination queue.
- The presence of an Advanced Message Security security policy.

This reason code can also occur in the Feedback field in the message descriptor of a report message; in this case it indicates that the error was encountered by a message channel agent when it attempted to put the message on a remote queue.

Completion code

MQCC_FAILED

Programmer response

Ensure that the correct queue manager or object is specified, and that appropriate authority exists.

If the application specifies a user ID and password, or an authentication token, when it connects, ensure that the credentials are valid.

Ask your system programmer or security administrator to check the queue manager logs for messages that provide additional information. Error messages that relate to authentication token problems might contain an error code that describes the cause of the error. For more information, see "Token authentication error codes" on page 1432.

This reason code is also used to identify the corresponding event message. The event message that is produced depends on the operation that failed, and is one of the following messages:

- MQCONN or MQCONNX "Not Authorized (type 1)" on page 196.
- MQOPEN or MQPUT1 "Not Authorized (type 2)" on page 198.
- MQCLOSE "Not Authorized (type 3)" on page 201.
- Command "Not Authorized (type 4)" on page 204.
- MQSUB "Not Authorized (type 5)" on page 205.
- MQSUB destination "Not Authorized (type 6)" on page 207.

Specific problems generating RC2035

JMSWMQ2013 invalid security authentication

See Invalid security authentication for information when your IBM MQ JMS application fails with security authentication errors.

MQRC_NOT_AUTHORIZED on a queue or channel

See MQRC_NOT_AUTHORIZED in WMQ for information when MQRC 2035 (MQRC_NOT_AUTHORIZED) is returned where a user is not authorized to perform the function. Determine which object the user cannot access and provide the user access to the object.

MQRC_NOT_AUTHORIZED (AMQ4036 on a client) as an administrator

See MQRC_NOT_AUTHORIZED as an administrator for information when MQRC 2035 (MQRC_NOT_AUTHORIZED) is returned where you try to use a user ID that is an IBM MQ Administrator, to remotely access the queue manager through a client connection.

MQS_REPORT_NOAUTH

See MQS_REPORT_NOAUTH for information on using this environment variable to better diagnose return code 2035 (MQRC_NOT_AUTHORIZED). The use of this environment variable generates errors in the queue manager error log, but does not generate a Failure Data Capture (FDC).

MQSAUTHERRORS

See MQSAUTHERRORS for information on using this environment variable to generate FDC files related to return code 2035 (MQRC_NOT_AUTHORIZED). The use of this environment variable generates an FDC, but does not generate errors in the queue manager error log.

Applications connecting to IBM MQ from WebSphere Application Server

See 2035 MQRC_NOT_AUTHORIZED when connecting to IBM MQ from WebSphere Application Server for information about troubleshooting MQRC 2035 (MQRC_NOT_AUTHORIZED) errors in an application that is connecting to IBM MQ from WebSphere Application Server.

Related reference

Return code= 2035 MQRC_NOT_AUTHORIZED

2036 (07F4) (RC2036): MQRC_NOT_OPEN_FOR_BROWSE

Explanation

An MQGET call was issued with one of the following options:

- MQGMO_BROWSE_FIRST
- MQGMO_BROWSE_NEXT
- MQGMO_BROWSE_MSG_UNDER_CURSOR
- MQGMO_MSG_UNDER_CURSOR

but either the queue had not been opened for browse, or you are using IBM MQ Multicast messaging.

Completion code

MQCC_FAILED

Programmer response

Specify MQOO_BROWSE when the queue is opened.

If you are using IBM MQ Multicast messaging, you cannot specify browse options with an MQGET call.

2037 (07F5) (RC2037): MQRC_NOT_OPEN_FOR_INPUT

Explanation

An MQGET call was issued to retrieve a message from a queue, but the queue had not been opened for input.

Completion code

MQCC_FAILED

Programmer response

Specify one of the following when the queue is opened:

- MQOO_INPUT_SHARED
- MQOO_INPUT_EXCLUSIVE
- MQOO_INPUT_AS_Q_DEF

2038 (07F6) (RC2038): MQRC_NOT_OPEN_FOR_INQUIRE

Explanation

An MQINQ call was issued to inquire object attributes, but the object had not been opened for inquire.

An MQINQ call was issued for a topic handle in IBM MQ Multicast.

Completion code

MQCC_FAILED

Programmer response

Specify MQOO_INQUIRE when the object is opened.

MQINQ is not supported for topic handles in IBM MQ Multicast.

2039 (07F7) (RC2039): MQRC_NOT_OPEN_FOR_OUTPUT

Explanation

An MQPUT call was issued to put a message on a queue, but the queue had not been opened for output.

Completion code

MQCC_FAILED

Programmer response

Specify MQOO_OUTPUT when the queue is opened.

2040 (07F8) (RC2040): MQRC_NOT_OPEN_FOR_SET

Explanation

An MQSET call was issued to set queue attributes, but the queue had not been opened for set.

An MQSET call was issued for a topic handle in IBM MQ Multicast.

Completion code

MQCC_FAILED

Programmer response

Specify MQOO_SET when the object is opened.

MQSET is not supported for topic handles in IBM MQ Multicast.

2041 (07F9) (RC2041): MQRC_OBJECT_CHANGED

Explanation

Object definitions that affect this object have been changed since the `Hobj` handle used on this call was returned by the MQOPEN call. For more information about the MQOPEN call, see MQOPEN.

This reason code can occur when running the REFRESH CLUSTER command. See Application issues seen when running REFRESH CLUSTER

This reason does not occur if the object handle is specified in the `Context` field of the **PutMsgOpts** parameter on the MQPUT or MQPUT1 call.

Completion code

MQCC_FAILED

Programmer response

Issue an MQCLOSE call to return the handle to the system. It is then usually sufficient to reopen the object and retry the operation. However, if the object definitions are critical to the application logic, an MQINQ call can be used after reopening the object, to obtain the new values of the object attributes.

2042 (07FA) (RC2042): MQRC_OBJECT_IN_USE

Explanation

An MQOPEN call was issued, but the object in question has already been opened by this or another application with options that conflict with those specified in the **Options** parameter. This arises if the request is for shared input, but the object is already open for exclusive input; it also arises if the request is for exclusive input, but the object is already open for input (of any sort).

MCAs for receiver channels, or the intra-group queuing agent (IGQ agent), may keep the destination queues open even when messages are not being transmitted; this results in the queues appearing to be *in use*. Use the MQSC command DISPLAY QSTATUS to find out who is keeping the queue open.

IBM MQ opens a queue for shared input if the application uses the MQOO_INPUT_SHARED open option or, if the application uses MQOO_INPUT_AS_Q_DEF and the default sharing option queue attribute is set to DEFSOPT(SHARED). However, there is an administrative override in the form of the SHARE/NOSHARE option.

If the queue definition shows NOSHARE, then IBM MQ will make the input handle exclusive regardless of the options set by the application.

- z/OS On z/OS, this reason can also occur for an MQOPEN or MQPUT1 call, if the object to be opened (which can be a queue, or for MQOPEN a namelist or process object) is in the process of being deleted.
- z/OS The default setting on z/OS is NOSHARE.

Completion code

MQCC_FAILED

Programmer response

System design should specify whether an application is to wait and retry, or take other action.

2043 (07FB) (RC2043): MQRC_OBJECT_TYPE_ERROR

Explanation

On the MQOPEN or MQPUT1 call, the `ObjectType` field in the object descriptor MQOD specifies a value that is not valid. For the MQPUT1 call, the object type must be MQOT_Q.

Completion code

MQCC_FAILED

Programmer response

Specify a valid object type.

2044 (07FC) (RC2044): MQRC_OD_ERROR

Explanation

On the MQOPEN or MQPUT1 call, the object descriptor MQOD is not valid, for one of the following reasons:

- The `StrucId` field is not MQOD_STRUC_ID.
- The `Version` field specifies a value that is not valid or not supported.
- The parameter pointer is not valid. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)
- The queue manager cannot copy the changed structure to application storage, even though the call is successful. This can occur, for example, if the pointer points to read-only storage.

Completion code

MQCC_FAILED

Programmer response

Ensure that input fields in the MQOD structure are set correctly.

2045 (07FD) (RC2045): MQRC_OPTION_NOT_VALID_FOR_TYPE

Explanation

On an MQOPEN or MQCLOSE call, an option is specified that is not valid for the type of object or queue being opened or closed.

For the MQOPEN call, this includes the following cases:

- An option that is inappropriate for the object type (for example, MQOO_OUTPUT for an MQOT_PROCESS object).
- An option that is unsupported for the queue type (for example, MQOO_INQUIRE for a remote queue that has no local definition).
- One or more of the following options:

- MQOO_INPUT_AS_Q_DEF
- MQOO_INPUT_SHARED
- MQOO_INPUT_EXCLUSIVE
- MQOO_BROWSE
- MQOO_INQUIRE
- MQOO_SET

when either:

- the queue name is resolved through a cell directory, or
- `ObjectQMgrName` in the object descriptor specifies the name of a local definition of a remote queue (to specify a queue manager alias), and the queue named in the `RemoteQMgrName` attribute of the definition is the name of the local queue manager.

For the MQCLOSE call, this includes the following case:

- The MQCO_DELETE or MQCO_DELETE_PURGE option when the queue is not a dynamic queue.

This reason code can also occur on the MQOPEN call when the object being opened is of type MQOT_NAMELIST, MQOT_PROCESS, or MQOT_Q_MGR, but the `ObjectQMgrName` field in MQOD is neither blank nor the name of the local queue manager.

Completion code

MQCC_FAILED

Programmer response

Specify the correct option. For the MQOPEN call, ensure that the `ObjectQMgrName` field is set correctly. For the MQCLOSE call, either correct the option or change the definition type of the model queue that is used to create the new queue.

2046 (07FE) (RC2046): MQRC_OPTIONS_ERROR

Explanation

The **`Options`** parameter or field contains options that are not valid, or a combination of options that is not valid.

- For the MQOPEN, MQCLOSE, MQXCNVC, mqBagToBuffer, mqBufferToBag, mqCreateBag, and mqExecute calls, **`Options`** is a separate parameter on the call.

  This reason also occurs if the parameter pointer is not valid. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)
- For the MQBEGIN, MQCONNX, MQGET, MQPUT, and MQPUT1 calls, `Options` is a field in the relevant options structure (MQBO, MQCNO, MQGMO, or MQPMO).
- For more information about option errors for IBM MQ Multicast see: MQI concepts and how they relate to multicast.

Completion code

MQCC_FAILED

Programmer response

Specify valid options. Check the description of the **`Options`** parameter or field to determine which options and combinations of options are valid. If multiple options are being set by adding the individual

options together, ensure that the same option is not added twice. For more information, see Rules for validating MQI options.

2047 (07FF) (RC2047): MQRC_PERSISTENCE_ERROR

Explanation

On an MQPUT or MQPUT1 call, the value specified for the `Persistence` field in the message descriptor MQMD is not valid.

Completion code

MQCC_FAILED

Programmer response

Specify one of the following values:

- MQPER_PERSISTENT
- MQPER_NOT_PERSISTENT
- MQPER_PERSISTENCE_AS_Q_DEF

2048 (0800) (RC2048): MQRC_PERSISTENT_NOT_ALLOWED

Explanation

On an MQPUT or MQPUT1 call, the value specified for the `Persistence` field in MQMD (or obtained from the `DefPersistence` queue attribute) specifies MQPER_PERSISTENT, but the queue on which the message is being placed does not support persistent messages. Persistent messages cannot be placed on temporary dynamic queues.

This reason code can also occur in the `Feedback` field in the message descriptor of a report message; in this case it indicates that the error was encountered by a message channel agent when it attempted to put the message on a remote queue.

Completion code

MQCC_FAILED

Programmer response

Specify MQPER_NOT_PERSISTENT if the message is to be placed on a temporary dynamic queue. If persistence is required, use a permanent dynamic queue or predefined queue in place of a temporary dynamic queue.

Be aware that server applications are recommended to send reply messages (message type MQMT_REPLY) with the same persistence as the original request message (message type MQMT_REQUEST). If the request message is persistent, the reply queue specified in the `ReplyToQ` field in the message descriptor MQMD cannot be a temporary dynamic queue. Use a permanent dynamic queue or predefined queue as the reply queue in this situation.

z/OS On z/OS, you cannot put persistent messages to a shared queue if the CFSTRUCT that the queue uses is defined with RECOVER(NO). Either put only non-persistent messages to this queue or change the CFSTRUCT definition to RECOVER(YES). If you put a persistent message to a queue that uses a CFSTRUCT with RECOVER(NO) the put will fail with MQRC_PERSISTENT_NOT_ALLOWED.

2049 (0801) (RC2049): MQRC_PRIORITY_EXCEEDS_MAXIMUM

Explanation

An MQPUT or MQPUT1 call was issued, but the value of the `Priority` field in the message descriptor MQMD exceeds the maximum priority supported by the local queue manager, as shown by the `MaxPriority` queue manager attribute. The message is accepted by the queue manager, but is placed on the queue at the queue manager's maximum priority. The `Priority` field in the message descriptor retains the value specified by the application that put the message.

Completion code

MQCC_WARNING

Programmer response

None required, unless this reason code was not expected by the application that put the message.

2050 (0802) (RC2050): MQRC_PRIORITY_ERROR

Explanation

An MQPUT or MQPUT1 call was issued, but the value of the `Priority` field in the message descriptor MQMD is not valid. The maximum priority supported by the queue manager is given by the `MaxPriority` queue manager attribute.

Completion code

MQCC_FAILED

Programmer response

Specify a value in the range zero through `MaxPriority`, or the special value MQPRI_PRIORITY_AS_Q_DEF.

2051 (0803) (RC2051): MQRC_PUT_INHIBITED

Explanation

MQPUT and MQPUT1 calls are currently inhibited for the queue, or for the queue to which this queue resolves.

This reason code can also occur in the `Feedback` field in the message descriptor of a report message; in this case it indicates that the error was encountered by a message channel agent when it attempted to put the message on a remote queue.

Completion code

MQCC_FAILED

Programmer response

If the system design allows put requests to be inhibited for short periods, retry the operation later.

This reason code is also used to identify the corresponding event message “Put Inhibited” on page 210.

System programmer action

Use ALTER QLOCAL(...) PUT(ENABLED) to allow messages to be put.

2052 (0804) (RC2052): MQRC_Q_DELETED

Explanation

An Hobj queue handle specified on a call refers to a dynamic queue that has been deleted since the queue was opened. For more information about the deletion of dynamic queues, see the description of MQCLOSE in MQCLOSE.

- z/OS On z/OS, this can also occur with the MQOPEN and MQPUT1 calls if a dynamic queue is being opened, but the queue is in a logically-deleted state. See MQCLOSE for more information about this.

Completion code

MQCC_FAILED

Programmer response

Issue an MQCLOSE call to return the handle and associated resources to the system (the MQCLOSE call will succeed in this case). Check the design of the application that caused the error.

2053 (0805) (RC2053): MQRC_Q_FULL

Explanation

An MQPUT or MQPUT1 call, or a command, failed because the queue is full, that is, it already contains the maximum number of messages possible, as specified by the **MaxQDepth** queue attribute.

This reason code can also occur in the Feedback field in the message descriptor of a report message; in this case it indicates that the error was encountered by a message channel agent when it attempted to put the message on a remote queue.

Completion code

MQCC_FAILED

Programmer response

Retry the operation later. Consider increasing the maximum depth for this queue, or arranging for more instances of the application to service the queue.

This reason code is also used to identify the corresponding event message “Queue Full” on page 215.

2055 (0807) (RC2055): MQRC_Q_NOT_EMPTY

Explanation

An MQCLOSE call was issued for a permanent dynamic queue, but the call failed because the queue is not empty or still in use. One of the following applies:

- The MQCO_DELETE option was specified, but there are messages on the queue.
- The MQCO_DELETE or MQCO_DELETE_PURGE option was specified, but there are uncommitted get or put calls outstanding against the queue.

See the usage notes pertaining to dynamic queues for the MQCLOSE call for more information.

This reason code is also returned from a command to clear or delete or move a queue, if the queue contains uncommitted messages (or committed messages in the case of delete queue without the purge option).

Completion code

MQCC_FAILED

Programmer response

Check why there might be messages on the queue. Be aware that the **CurrentQDepth** queue attribute might be zero even though there are one or more messages on the queue; this can happen if the messages have been retrieved as part of a unit of work that has not yet been committed. If the messages can be discarded, try using the MQCLOSE call with the MQCO_DELETE_PURGE option. Consider retrying the call later.

2056 (0808) (RC2056): MQRC_Q_SPACE_NOT_AVAILABLE

Explanation

An MQPUT or MQPUT1 call was issued, but there is no space available for the queue on disk or other storage device.

This reason code can also occur in the Feedback field in the message descriptor of a report message; in this case it indicates that the error was encountered by a message channel agent when it attempted to put the message on a remote queue.

- z/OS On z/OS, this reason code does not occur.

Completion code

MQCC_FAILED

Programmer response

Check whether an application is putting messages in an infinite loop. If not, make more disk space available for the queue.

2057 (0809) (RC2057): MQRC_Q_TYPE_ERROR

Explanation

One of the following occurred:

- On an MQOPEN call, the ObjectQMgrName field in the object descriptor MQOD or object record MQOR specifies the name of a local definition of a remote queue (to specify a queue manager alias), and in that local definition the **RemoteQMgrName** attribute is the name of the local queue manager. However, the ObjectName field in MQOD or MQOR specifies the name of a model queue on the local queue manager; this is not allowed. For more information, see MQOPEN.
- If the environment variable AMQ_BLOCK_RECONN_DYN_QUEUES has been set on a queue manager in a uniform cluster, this return code can occur on an MQOPEN call when the call results in the creation of a new dynamic queue, if the connection to the queue manager was made using the connect option MQCNO_RECONNECT.
- On an MQPUT1 call, the object descriptor MQOD or object record MQOR specifies the name of a model queue.

- On a previous MQPUT or MQPUT1 call, the ReplyToQ field in the message descriptor specified the name of a model queue, but a model queue cannot be specified as the destination for reply or report messages. Only the name of a predefined queue, or the name of the dynamic queue created from the model queue, can be specified as the destination. In this situation the reason code MQRC_Q_TYPE_ERROR is returned in the Reason field of the MQDLH structure when the reply message or report message is placed on the dead-letter queue.

Completion code

MQCC_FAILED

Programmer response

Specify a valid queue.

This reason code is also used to identify the corresponding event message "Queue Type Error" on page 221.

2058 (080A) (RC2058): MQRC_Q_MGR_NAME_ERROR

Explanation

On an MQCONN or MQCONNX call, the value specified for the **QMgrName** parameter is not valid or not known. This reason also occurs if the parameter pointer is not valid. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)

- z/OS On z/OS for CICS applications, this reason can occur on any call if the original connect specified an incorrect or unrecognized name.

z/OS For CICS, this reason can be caused by the wrong resync value. For example, Groupresync is specified and the queue manager is not in a queue sharing group.

This reason code can also occur if an MQ MQI client application attempts to connect to a queue manager within an MQ-client queue manager group (see the **QMgrName** parameter of MQCONN), and either:

- Queue manager groups are not supported.
- There is no queue manager group with the specified name.

z/OS For the IMS adapter on z/OS, MQRC_Q_MGR_NAME_ERROR occurs when CSQQDEFV is loaded and one of:

- MQCONN call specified a blank queue manager name, and there is no CSQQDEFX TYPE=DEFAULT entry.
- Queue manager name specified for the MQCONN call is not found in the loaded CSQQDEFV table.
- A different CSQQDEFV module than the one intended has been loaded, such as the default one shipped with the product. This situation can be caused by the library containing your CSQQDEFV module being after the SCSQAUTH data set in STEPLIB or JOBLIB for the dependent region. Move your library before the SCSQAUTH data set in the concatenation. If there is not a CSQQDEFV load module in STEPLIB/JOBLIB, check what is loaded through the LINKLIST.

z/OS For the CICS adapter, MQRC_Q_MGR_NAME_ERROR occurs when:

- The MQCONN definition specifies Resyncmember(Groupresync) and the MQNAME specifies a queue manager name instead of a queue sharing group name.

z/OS MQRC_Q_MGR_NAME_ERROR is returned when an attempt is made to connect the same address space using a different type of adapter from that used by an existing IBM MQ connection. Specific cases include:

- An exit running in a CICS, IMS or RRSBATCH type region, attempts an MQCONN via the BATCH adapter.

- A CICS Standard Liberty profile application attempts to connect to IBM MQ with bindings connection factory and there is already an existing CICS MQCONN established from the CICS region to the same queue manager.

Completion code

MQCC_WARNING or MQCC_FAILED

Programmer response

Use an all-blank name if possible, or verify that the name used is valid.

If you are using CICS Resyncmember(Groupresync), use the queue sharing group (QSG) name in the MQNAME rather than the queue manager name.

2059 (080B) (RC2059): MQRC_Q_MGR_NOT_AVAILABLE

Explanation

This error occurs:

1. On an MQCONN or MQCONNX call, the queue manager identified by the **QMgrName** parameter is not available for connection.
   - z/OS On z/OS:
     - For batch applications, this reason can be returned to applications running in LPARs that do not have a queue manager installed.
     - For CICS applications, this reason can occur on any call if the original connect specified a queue manager with a name that was recognized, but which is not available.
     - There might be a system wide ECSA shortage, or the storage available for creating new queue manager connections might be limited by the ACELIM system parameter. See message CSQ3202E for more information.
   - IBM i On IBM i, this reason can also be returned by the MQOPEN and MQPUT1 calls, when MQHC_DEF_HCONN is specified for the **Hconn** parameter by an application running in compatibility mode.
2. On an MQCONN or MQCONNX call from an IBM MQ MQI client application:
   - Attempting to connect to a queue manager within an MQ-client queue manager group when none of the queue managers in the group is available for connection (see the **QMgrName** parameter of the MQCONN call).
   - If the client channel fails to connect, perhaps because of an error with the client-connection or the corresponding server-connection channel definitions.
3. If a command uses the **CommandScope** parameter specifying a queue manager that is not active in the queue sharing group.

Completion code

MQCC_FAILED

Programmer response

Ensure that the queue manager has been started. If the connection is from a client application, check the channel definitions, channel status, and error logs.

In a multiple installation environment, ensure that IBM MQ libraries are loaded by the operating system. For more information, see Connecting applications in a multiple installation environment.

2061 (080D) (RC2061): MQRC_REPORT_OPTIONS_ERROR

Explanation

An MQPUT or MQPUT1 call was issued, but the `Report` field in the message descriptor MQMD contains one or more options that are not recognized by the local queue manager. The options that cause this reason code to be returned depend on the destination of the message; see the description of REPORT in Report options and message flags for more details.

This reason code can also occur in the `Feedback` field in the MQMD of a report message, or in the `Reason` field in the MQDLH structure of a message on the dead-letter queue; in both cases it indicates that the destination queue manager does not support one or more of the report options specified by the sender of the message.

Completion code

MQCC_FAILED

Programmer response

Do the following:

- Ensure that the `Report` field in the message descriptor is initialized with a value when the message descriptor is declared, or is assigned a value prior to the MQPUT or MQPUT1 call. Specify MQRO_NONE if no report options are required.
- Ensure that the report options specified are valid; see the `Report` field described in the description of MQMD in Report options and message flags for valid report options.
- If multiple report options are being set by adding the individual report options together, ensure that the same report option is not added twice.
- Check that conflicting report options are not specified. For example, do not add both MQRO_EXCEPTION and MQRO_EXCEPTION_WITH_DATA to the `Report` field; only one of these can be specified.

z/OS 2062 (080E) (RC2062): MQRC_SECOND_MARK_NOT_ALLOWED

Explanation

An MQGET call was issued specifying the MQGMO_MARK_SKIP_BACKOUT option in the `Options` field of MQGMO, but a message has already been marked within the current unit of work. Only one marked message is allowed within each unit of work.

This reason code occurs only on z/OS.

Completion code

MQCC_FAILED

Programmer response

Modify the application so that no more than one message is marked within each unit of work.

2063 (080F) (RC2063): MQRC_SECURITY_ERROR

Explanation

An MQCONN, MQCONNX, MQOPEN, MQSUB, MQPUT1, or MQCLOSE call was issued, but it failed because a security error occurred.

- z/OS On z/OS, there are two possible reasons for this:
  - An MQCONN or MQCONNX call was issued to connect to the queue manager using the BINDINGS transport from a JMS application running inside a WebSphere Application Server, or WebSphere Liberty Profile environment, passing in a username or password, or both, that were longer than 8 characters.
  - The security error was returned by the External Security Manager.
- If you are using Advanced Message Security (AMS), this might be a set up issue.

  z/OS An IBM MQ API call was issued while AMS was enabled, but the call failed because a security error occurred during AMS processing.

  z/OS An MQOPEN call might fail if a valid certificate does not exist, for example.

  z/OS An MQGET call might fail due to certificates or policies being configured incorrectly, for example. For a failing MQGET call, messages might be delivered to the SYSTEM.PROTECTION.ERROR.QUEUE.
- If you are using connection authentication with an LDAP server, this might be as a result of connectivity failure to the LDAP server, or an error from the LDAP server.
- If the application supplied an authentication token on an MQCONNX call, there might be an error in the queue manager authentication token configuration.

Completion code

MQCC_FAILED

Programmer response

Note the error from the security manager, and contact your system programmer or security administrator.

- If you are using Advanced Message Security, check the queue manager error logs.
- z/OS On z/OS, ensure that the username specified has a maximum length of 8 characters, and the password specified has a maximum length of 100 characters.

  Ask your system programmer or security administrator to:
  - Check the queue manager and AMS job logs for additional messages
  - Verify that certificates are valid and have been correctly configured
  - Confirm that policies are valid and also correctly configured
  - Check for any messages on the SYSTEM.PROTECTION.ERROR.QUEUE.
- IBM i On IBM i, the FFST log will contain the error information.
- If you are using LDAP, use the DISPLAY QMSTATUS command to check the status of the connection to the LDAP server, and check the queue manager error logs for any error messages.
- If the application supplied an authentication token on an MQCONNX call, ask your system programmer or security administrator to check the queue manager error logs for messages that provide additional information. Error messages that relate to authentication token problems might contain an error code that describes the cause of the error. For more information, see “Token authentication error codes” on page 1432.

Linux AIX 2064 (0810) (RC2064): MQRC_TOKEN_TIMESTAMP_NOT_VALID

Explanation

An MQCONNX call was issued that included an authentication token, but it failed because the token is not yet valid or is expired.

Completion code

MQCC_FAILED

Programmer response

Ask your administrator to check the queue manager error logs for additional messages. Get another token that is not expired and if it has a 'not before' claim, connect and present the token after the 'not before' time is passed. For more information, see Requirements for authentication tokens.

Related concepts

MQCONNX - Connect queue manager (extended)

Working with authentication tokens

Troubleshooting authentication token problems

Related tasks

Using authentication tokens in an application

Related reference

Token authentication error codes

2065 (0811) (RC2065): MQRC_SELECTOR_COUNT_ERROR

Explanation

On an MQINQ or MQSET call, the **SelectorCount** parameter specifies a value that is not valid. This reason also occurs if the parameter pointer is not valid. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)

Completion code

MQCC_FAILED

Programmer response

Specify a value in the range 0 through 256.

2066 (0812) (RC2066): MQRC_SELECTOR_LIMIT_EXCEEDED

Explanation

On an MQINQ or MQSET call, the **SelectorCount** parameter specifies a value that is larger than the maximum supported (256).

Completion code

MQCC_FAILED

Programmer response

Reduce the number of selectors specified on the call; the valid range is 0 through 256.

2067 (0813) (RC2067): MQRC_SELECTOR_ERROR

Explanation

An MQINQ or MQSET call was issued, but the `Selectors` array contains a selector that is not valid for one of the following reasons:

- The selector is not supported or out of range.
- The selector is not applicable to the type of object with attributes that are being inquired upon or set.
- The selector is for an attribute that cannot be set.

This reason also occurs if the parameter pointer is not valid. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)

An MQINQ call was issued for a managed handle in IBM MQ Multicast, inquiring a value other than **`Current Depth`**.

Completion code

MQCC_FAILED

Programmer response

Ensure that the value specified for the selector is valid for the object type represented by `Hobj`. For the MQSET call, also ensure that the selector represents an integer attribute that can be set.

MQINQ for managed handles in IBM MQ Multicast can only inquire on *`Current Depth`*.

2068 (0814) (RC2068): MQRC_SELECTOR_NOT_FOR_TYPE

Explanation

On the MQINQ call, one or more selectors in the `Selectors` array is not applicable to the type of the queue with attributes that are being inquired upon.

This reason also occurs when the queue is a cluster queue that resolved to a remote instance of the queue. In this case only a subset of the attributes that are valid for local queues can be inquired. See the usage notes in the description of MQINQ in MQINQ - Inquire object attributes for more information about MQINQ.

The call completes with MQCC_WARNING, with the attribute values for the inapplicable selectors set as follows:

- For integer attributes, the corresponding elements of **`IntAttrs`** are set to MQIAV_NOT_APPLICABLE.
- For character attributes, the appropriate parts of the **`CharAttrs`** string are set to a character string consisting entirely of asterisks (*).

Completion code

MQCC_WARNING

Programmer response

Verify that the selector specified is the one that was intended.

If the queue is a cluster queue, specifying one of the MQOO_BROWSE, MQOO_INPUT_*, or MQOO_SET options in addition to MQOO_INQUIRE forces the queue to resolve to the local instance of the queue. However, if there is no local instance of the queue the MQOPEN call fails.

Windows z/OS 2069 (0815) (RC2069): MQRC_SIGNAL_OUTSTANDING

Explanation

An MQGET call was issued with either the MQGMO_SET_SIGNAL or MQGMO_WAIT option, but there is already a signal outstanding for the queue handle *Hobj*.

This reason code occurs only in z/OS and Windows.

Completion code

MQCC_FAILED

Programmer response

Check the application logic. If it is necessary to set a signal or wait when there is a signal outstanding for the same queue, a different object handle must be used.

Windows z/OS 2070 (0816) (RC2070): MQRC_SIGNAL_REQUEST_ACCEPTED

Explanation

An MQGET call was issued specifying MQGMO_SET_SIGNAL in the **GetMsgOpts** parameter, but no suitable message was available; the call returns immediately. The application can now wait for the signal to be delivered.

- z/OS On z/OS, the application should wait on the Event Control Block pointed to by the *Signal1* field.
- Windows On Windows, the application should wait for the signal Windows message to be delivered.

This reason code occurs only in z/OS and Windows.

Completion code

MQCC_WARNING

Programmer response

Wait for the signal; when it is delivered, check the signal to ensure that a message is now available. If it is, reissue the MQGET call.

- On z/OS, wait on the ECB pointed to by the *Signal1* field and, when it is posted, check it to ensure that a message is now available.
- On Windows 95, Windows 98, the application (thread) should continue executing its message loop.

2071 (0817) (RC2071): MQRC_STORAGE_NOT_AVAILABLE

Explanation

The call failed because there is insufficient main storage available.

Completion code

MQCC_FAILED

Programmer response

Ensure that active applications are behaving correctly, for example, that they are not looping unexpectedly. If no problems are found, make more main storage available.

- z/OS On z/OS, if no application problems are found, ask your system programmer to increase the size of the region in which the queue manager runs.
- z/OS On z/OS, this return code can be seen as a result of exhausting all storage in the application address space.

2072 (0818) (RC2072): MQRC_SYNCPOINT_NOT_AVAILABLE

Explanation

Either the MQGMO_SYNCPOINT option was used with an MQGET call, or the MQPMO_SYNCPOINT option was used with an MQPUT or MQPUT1 call, but the local queue manager was unable to honor the request. If the queue manager does not support units of work, the **SyncPoint** queue manager attribute has the value MQSP_NOT_AVAILABLE.

This reason code can also occur on the MQGET, MQPUT, and MQPUT1 calls when an external unit-of-work coordinator is used. If that coordinator requires an explicit call to start the unit of work, but the application has not issued that call before the MQGET, MQPUT, or MQPUT1 call, reason code MQRC_SYNCPOINT_NOT_AVAILABLE is returned.

IBM i On IBM i, this reason code means that IBM i Commitment Control is not started, or is unavailable for use by the queue manager.

This reason code can also be returned if the MQGMO_SYNCPOINT or the MQPMO_SYNCPOINT option was used for IBM MQ Multicast messaging. Transactions are not supported for multicast.

Completion code

MQCC_FAILED

Programmer response

Remove the specification of MQGMO_SYNCPOINT or MQPMO_SYNCPOINT, as appropriate.

IBM i On IBM i, ensure that Commitment Control is started. If this reason code occurs after Commitment Control is started, contact your system programmer.

2075 (081B) (RC2075): MQRC_TRIGGER_CONTROL_ERROR

Explanation

On an MQSET call, the value specified for the MQIA_TRIGGER_CONTROL attribute selector is not valid.

Completion code

MQCC_FAILED

Programmer response

Specify a valid value.

2076 (081C) (RC2076): MQRC_TRIGGER_DEPTH_ERROR

Explanation

On an MQSET call, the value specified for the MQIA_TRIGGER_DEPTH attribute selector is not valid.

Completion code

MQCC_FAILED

Programmer response

Specify a value that is greater than zero.

2077 (081D) (RC2077): MQRC_TRIGGER_MSG_PRIORITY_ERR

Explanation

On an MQSET call, the value specified for the MQIA_TRIGGER_MSG_PRIORITY attribute selector is not valid.

Completion code

MQCC_FAILED

Programmer response

Specify a value in the range zero through the value of **MaxPriority** queue manager attribute.

2078 (081E) (RC2078): MQRC_TRIGGER_TYPE_ERROR

Explanation

On an MQSET call, the value specified for the MQIA_TRIGGER_TYPE attribute selector is not valid.

Completion code

MQCC_FAILED

Programmer response

Specify a valid value.

2079 (081F) (RC2079): MQRC_TRUNCATED_MSG_ACCEPTED

Explanation

On an MQGET call, the message length was too large to fit into the supplied buffer. The MQGMO_ACCEPT_TRUNCATED_MSG option was specified, so the call completes. The message is removed from the queue (subject to unit-of-work considerations), or, if this was a browse operation, the browse cursor is advanced to this message.

The **DataLength** parameter is set to the length of the message before truncation, the **Buffer** parameter contains as much of the message as fits, and the MQMD structure is filled in.

Completion code

MQCC_WARNING

Programmer response

None, because the application expected this situation.

2080 (0820) (RC2080): MQRC_TRUNCATED_MSG_FAILED

Explanation

On an MQGET call, the message length was too large to fit into the supplied buffer. The MQGMO_ACCEPT_TRUNCATED_MSG option was not specified, so the message has not been removed from the queue. If this was a browse operation, the browse cursor remains where it was before this call, but if MQGMO_BROWSE_FIRST was specified, the browse cursor is positioned logically before the highest-priority message on the queue.

The `DataLength` field is set to the length of the message before truncation, the **Buffer** parameter contains as much of the message as fits, and the MQMD structure is filled in.

Completion code

MQCC_WARNING

Programmer response

Supply a buffer that is at least as large as `DataLength`, or specify MQGMO_ACCEPT_TRUNCATED_MSG if not all of the message data is required.

2082 (0822) (RC2082): MQRC_UNKNOWN_ALIAS_BASE_Q

Explanation

An MQOPEN or MQPUT1 call was issued specifying an alias queue as the target, but the **BaseQName** in the alias queue attributes is not recognized as a queue name.

This reason code can occur when **BaseQName** is the name of a cluster queue that cannot be resolved successfully.

This reason code can also occur when running the REFRESH CLUSTER command. See Application issues seen when running REFRESH CLUSTER

MQRC_UNKNOWN_ALIAS_BASE_Q might indicate that the application is specifying the **ObjectQmgrName** of the queue manager that it is connecting to, and the queue manager that is hosting the alias queue. This means that the queue manager looks for the alias target queue on the specified queue manager and fails because the alias target queue is not on the local queue manager. Leave the **ObjectQmgrName** parameter blank so that the clustering decides which queue manager to route to.

Completion code

MQCC_FAILED

Programmer response

Correct the queue definitions.

This reason code is also used to identify the corresponding event message "Unknown Alias Base Queue" on page 234.

If the reason code is seen by an application that is using IBM MQ classes for JMS, modify the JMS queue object definition that is used by the application so that the **QMANAGER** property is set to the empty string (""). This setting ensures that the clustering decides which queue manager to route to.

If the queue is in the cluster, check that you have used appropriate open options. You cannot get messages from a remote cluster queue, so make sure that the open options are for output only.

Related reference

Return code=2082 MQRC_UNKNOWN_ALIAS_BASE_Q opening a queue in the cluster

2085 (0825) (RC2085): MQRC_UNKNOWN_OBJECT_NAME

Explanation

An MQOPEN, MQPUT1 , or MQSUB call was issued, but the object identified by the `ObjectName` and `ObjectQMgrName` fields in the object descriptor MQOD cannot be found. One of the following applies:

- The `ObjectQMgrName` field is one of the following:
  - Blank
  - The name of the local queue manager
  - The name of a local definition of a remote queue (a queue manager alias) in which the **RemoteQMgrName** attribute is the name of the local queue manager

  but no object with the specified `ObjectName` and `ObjectType` exists on the local queue manager.
- The object being opened is a cluster queue that is hosted on a remote queue manager, but the local queue manager does not have a defined route to the remote queue manager.
- The object being opened is a queue definition that has QSGDISP(GROUP). Such definitions cannot be used with the MQOPEN, MQPUT1 , or MQSUB calls.
- The MQOD in the failing application specifies the name of the local queue manager in `ObjectQMgrName`. The local queue manager does not host the particular cluster queue specified in `ObjectName`.

  The solution in this environment is to leave `ObjectQMgrName` of the MQOD blank.

This reason code can occur when running the REFRESH CLUSTER command. See Application issues seen when running REFRESH CLUSTER

This reason code can also occur in response to a command that specifies the name of an object or other item that does not exist.

Completion code

MQCC_FAILED

Programmer response

Specify a valid object name. Ensure that the name is padded with blanks at the end, if necessary. If this is correct, check the object definitions.

This reason code is also used to identify the corresponding event message "Unknown Object Name" on page 238.

If the queue is in the cluster, check that you have used appropriate open options. You cannot get messages from a remote cluster queue, so make sure that the open options are for output only.

Related reference

Return code= 2085 MQRC_UNKNOWN_OBJECT_NAME when trying to open a queue in the cluster

2086 (0826) (RC2086): MQRC_UNKNOWN_OBJECT_Q_MGR

Explanation

On an MQOPEN or MQPUT1 call, the ObjectQMgrName field in the object descriptor MQOD does not satisfy the naming rules for objects. For more information, see ObjectQMgrName (MQCHAR48).

This reason also occurs if the ObjectType field in the object descriptor has the value MQOT_Q_MGR, and the ObjectQMgrName field is not blank, but the name specified is not the name of the local queue manager.

Completion code

MQCC_FAILED

Programmer response

Specify a valid queue manager name. To refer to the local queue manager, a name consisting entirely of blanks or beginning with a null character can be used. Ensure that the name is padded with blanks at the end, or terminated with a null character if necessary.

2087 (0827) (RC2087): MQRC_UNKNOWN_REMOTE_Q_MGR

Explanation

On an MQOPEN or MQPUT1 call, an error occurred with the queue-name resolution, for one of the following reasons:

- ObjectQMgrName is blank or the name of the local queue manager, ObjectName is the name of a local definition of a remote queue (or an alias to one), and one of the following is true:
  - RemoteQMgrName is blank or the name of the local queue manager. Note that this error occurs even if XmitQName is not blank.
  - XmitQName is blank, but there is no transmission queue defined with the name of RemoteQMgrName, and the **DefXmitQName** queue manager attribute is blank.
  - RemoteQMgrName and RemoteQName specify a cluster queue that cannot be resolved successfully, and the **DefXmitQName** queue manager attribute is blank.
  - z/OS On z/OS only, the RemoteQMgrName is the name of a queue manager in the queue sharing group but intra-group queuing is disabled.
- ObjectQMgrName is the name of a local definition of a remote queue (containing a queue manager alias definition), and one of the following is true:
  - RemoteQName is not blank.
  - XmitQName is blank, but there is no transmission queue defined with the name of RemoteQMgrName, and the **DefXmitQName** queue manager attribute is blank.
- ObjectQMgrName is not:
  - Blank
  - The name of the local queue manager
  - The name of a transmission queue

– The name of a queue manager alias definition (that is, a local definition of a remote queue with a blank RemoteQName)

but the **DefXmitQName** queue manager attribute is blank and the queue manager is not part of a queue sharing group with intra-group queuing enabled.

- ObjectQMgrName is the name of a model queue.
- The queue name is resolved through a cell directory. However, there is no queue defined with the same name as the remote queue manager name obtained from the cell directory, and the DefXmitQName queue manager attribute is blank.

Completion code

MQCC_FAILED

Programmer response

Check the values specified for ObjectQMgrName and ObjectName. If these are correct, check the queue definitions.

This reason code is also used to identify the corresponding event message "Unknown Remote Queue Manager" on page 240.

2090 (082A) (RC2090): MQRC_WAIT_INTERVAL_ERROR

Explanation

On the MQGET call, the value specified for the WaitInterval field in the **GetMsgOpts** parameter is not valid.

Completion code

MQCC_FAILED

Programmer response

Specify a value greater than or equal to zero, or the special value MQWI_UNLIMITED if an indefinite wait is required.

2091 (082B) (RC2091): MQRC_XMIT_Q_TYPE_ERROR

Explanation

On an MQOPEN or MQPUT1 call, a message is to be sent to a remote queue manager. The ObjectName or ObjectQMgrName field in the object descriptor specifies the name of a local definition of a remote queue but one of the following applies to the XmitQName attribute of the definition:

- XmitQName is not blank, but specifies a queue that is not a local queue
- XmitQName is blank, but RemoteQMgrName specifies a queue that is not a local queue

This reason also occurs if the queue name is resolved through a cell directory, and the remote queue manager name obtained from the cell directory is the name of a queue, but this is not a local queue.

Completion code

MQCC_FAILED

Programmer response

Check the values specified for `ObjectName` and `ObjectQMgrName`. If these are correct, check the queue definitions.

This reason code is also used to identify the corresponding event message "Transmission Queue Type Error" on page 230.

2092 (082C) (RC2092): MQRC_XMIT_Q_USAGE_ERROR

Explanation

On an MQOPEN or MQPUT1 call, a message is to be sent to a remote queue manager, but one of the following occurred:

- `ObjectQMgrName` specifies the name of a local queue, but it does not have a `Usage` attribute of MQUS_TRANSMISSION.
- The `ObjectName` or `ObjectQMgrName` field in the object descriptor specifies the name of a local definition of a remote queue but one of the following applies to the **XmitQName** attribute of the definition:
  - **XmitQName** is not blank, but specifies a queue that does not have a **Usage** attribute of MQUS_TRANSMISSION
  - **XmitQName** is blank, but `RemoteQMgrName` specifies a queue that does not have a **Usage** attribute of MQUS_TRANSMISSION
  - *XmitQName* specifies the queue SYSTEM.QSG.TRANSMIT.QUEUE the IGQ queue manager attribute indicates that IGQ is DISABLED.
- The queue name is resolved through a cell directory, and the remote queue manager name obtained from the cell directory is the name of a local queue, but it does not have a **Usage** attribute of MQUS_TRANSMISSION.

Completion code

MQCC_FAILED

Programmer response

Check the values specified for `ObjectName` and `ObjectQMgrName`. If these are correct, check the queue definitions.

This reason code is also used to identify the corresponding event message "Transmission Queue Usage Error" on page 232.

2093 (082D) (RC2093): MQRC_NOT_OPEN_FOR_PASS_ALL

Explanation

An MQPUT call was issued with the MQPMO_PASS_ALL_CONTEXT option specified in the **PutMsgOpts** parameter, but the queue had not been opened with the MQOO_PASS_ALL_CONTEXT option.

Completion code

MQCC_FAILED

Programmer response

Specify MQOO_PASS_ALL_CONTEXT (or another option that implies it) when the queue is opened.

2094 (082E) (RC2094): MQRC_NOT_OPEN_FOR_PASS_IDENT

Explanation

An MQPUT call was issued with the MQPMO_PASS_IDENTITY_CONTEXT option specified in the **PutMsgOpts** parameter, but the queue had not been opened with the MQOO_PASS_IDENTITY_CONTEXT option.

Completion code

MQCC_FAILED

Programmer response

Specify MQOO_PASS_IDENTITY_CONTEXT (or another option that implies it) when the queue is opened.

2095 (082F) (RC2095): MQRC_NOT_OPEN_FOR_SET_ALL

Explanation

An MQPUT call was issued with the MQPMO_SET_ALL_CONTEXT option specified in the **PutMsgOpts** parameter, but the queue had not been opened with the MQOO_SET_ALL_CONTEXT option.

Completion code

MQCC_FAILED

Programmer response

Specify MQOO_SET_ALL_CONTEXT when the queue is opened.

2096 (0830) (RC2096): MQRC_NOT_OPEN_FOR_SET_IDENT

Explanation

An MQPUT call was issued with the MQPMO_SET_IDENTITY_CONTEXT option specified in the **PutMsgOpts** parameter, but the queue had not been opened with the MQOO_SET_IDENTITY_CONTEXT option.

Completion code

MQCC_FAILED

Programmer response

Specify MQOO_SET_IDENTITY_CONTEXT (or another option that implies it) when the queue is opened.

2097 (0831) (RC2097): MQRC_CONTEXT_HANDLE_ERROR

Explanation

On an MQPUT or MQPUT1 call, MQPMO_PASS_IDENTITY_CONTEXT or MQPMO_PASS_ALL_CONTEXT was specified, but the handle specified in the `Context` field of the **PutMsgOpts** parameter is either not a valid queue handle, or it is a valid queue handle but the queue was not opened with MQOO_SAVE_ALL_CONTEXT.

Completion code

MQCC_FAILED

Programmer response

Specify MQOO_SAVE_ALL_CONTEXT when the queue referred to is opened.

2098 (0832) (RC2098): MQRC_CONTEXT_NOT_AVAILABLE

Explanation

On an MQPUT or MQPUT1 call, MQPMO_PASS_IDENTITY_CONTEXT or MQPMO_PASS_ALL_CONTEXT was specified, but the queue handle specified in the `Context` field of the **PutMsgOpts** parameter has no context associated with it. This arises if no message has yet been successfully retrieved with the queue handle referred to, or if the last successful MQGET call was a browse.

This condition does not arise if the message that was last retrieved had no context associated with it.

- z/OS On z/OS, if a message is received by a message channel agent that is putting messages with the authority of the user identifier in the message, this code is returned in the *Feedback* field of an exception report if the message has no context associated with it.

Completion code

MQCC_FAILED

Programmer response

Ensure that a successful nonbrowse get call has been issued with the queue handle referred to.

Windows z/OS 2099 (0833) (RC2099): MQRC_SIGNAL1_ERROR

Explanation

An MQGET call was issued, specifying MQGMO_SET_SIGNAL in the **GetMsgOpts** parameter, but the `Signal1` field is not valid.

- z/OS On z/OS, the address contained in the `Signal1` field is not valid, or points to read-only storage. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)
- Windows On Windows, the window handle in the `Signal1` field is not valid.

This reason code occurs only on z/OSand Windows .

Completion code

MQCC_FAILED

Programmer response

Correct the setting of the `Signal1` field.

2100 (0834) (RC2100): MQRC_OBJECT_ALREADY_EXISTS

Explanation

An MQOPEN call was issued to create a dynamic queue, but a queue with the same name as the dynamic queue already exists.

- z/OS On z/OS, a rare *race condition* can also give rise to this reason code; see the description of reason code MQRC_NAME_IN_USE for more details.

Completion code

MQCC_FAILED

Programmer response

If supplying a dynamic queue name in full, ensure that it obeys the naming conventions for dynamic queues; if it does, either supply a different name, or delete the existing queue if it is no longer required. Alternatively, allow the queue manager to generate the name.

If the queue manager is generating the name (either in part or in full), reissue the MQOPEN call.

2101 (0835) (RC2101): MQRC_OBJECT_DAMAGED

Explanation

The object accessed by the call is damaged and cannot be used. For example, this might be because the definition of the object in main storage is not consistent, or because it differs from the definition of the object on disk, or because the definition on disk cannot be read. The object can be deleted, although it might not be possible to delete the associated user space.

- z/OS On z/OS, this reason occurs when the Db2 list header or structure number associated with a shared queue is zero. This situation arises as a result of using the MQSC command DELETE CFSTRUCT to delete the Db2 structure definition. The command resets the list header and structure number to zero for each of the shared queues that references the deleted CF structure.

Completion code

MQCC_FAILED

Programmer response

It might be necessary to stop and restart the queue manager, or to restore the queue manager data from backup storage.

- On the following platforms, consult the FFST record to obtain more detail about the problem:
  - Linux AIX AIX and Linux
  - IBM i IBM i
- z/OS On z/OS, delete the shared queue and redefine it using the MQSC command DEFINE QLOCAL. This automatically defines a CF structure and allocates list headers for it.

OpenShift Native HA queue managers, which use replicated logging, automatically attempt asynchronous media recovery of damaged objects that are configured as media recoverable. If the automatic recovery is initially unsuccessful, the asynchronous recovery processing periodically retries. If the issue that prevented recovery can be resolved, the object is recovered on the next retry, or the

object can be recovered manually, using the **rcrmqobj** command. For circular logging queue managers, or for objects that are not configured as media recoverable, no media recovery is possible, so the object should be deleted.

2102 (0836) (RC2102): MQRC_RESOURCE_PROBLEM

Explanation

There are insufficient system resources to complete the call successfully.

z/OS On z/OS this can indicate that Db2 errors occurred when using shared queues, or that the maximum number of shared queues that can be defined in a single coupling facility list structure has been reached.

Completion code

MQCC_FAILED

Programmer response

Run the application when the machine is less heavily loaded.

- z/OS On z/OS, check the operator console for messages that might provide additional information.
- On the following platforms, consult the FFST record to obtain more detail about the problem:
  - IBM i IBM i
  - Linux AIX AIX and Linux

Multi 2103 (0837) (RC2103): MQRC_ANOTHER_Q_MGR_CONNECTED

Explanation

An MQCONN or MQCONNX call was issued, but the thread or process is already connected to a different queue manager. The thread or process can connect to only one queue manager at a time.

- z/OS On z/OS, this reason code does not occur.
- Windows On Windows, MTS objects do not receive this reason code, as connections to other queue managers are allowed.

Completion code

MQCC_FAILED

Programmer response

Use the MQDISC call to disconnect from the queue manager that is already connected, and then issue the MQCONN or MQCONNX call to connect to the new queue manager.

Disconnecting from the existing queue manager closes any queues that are currently open; it is suggested that any uncommitted units of work are committed or backed out before the MQDISC call is issued.

2104 (0838) (RC2104): MQRC_UNKNOWN_REPORT_OPTION

Explanation

An MQPUT or MQPUT1 call was issued, but the `Report` field in the message descriptor MQMD contains one or more options that are not recognized by the local queue manager. The options are accepted.

The options that cause this reason code to be returned depend on the destination of the message; see the description of REPORT in Report options and message flags for more information.

Completion code

MQCC_WARNING

Programmer response

If this reason code is expected, no corrective action is required. If this reason code is not expected, do the following:

- Ensure that the `Report` field in the message descriptor is initialized with a value when the message descriptor is declared, or is assigned a value prior to the MQPUT or MQPUT1 call.
- Ensure that the report options specified are valid; see the `Report` field described in the description of MQMD in MQMD - Message descriptor for valid report options.
- If multiple report options are being set by adding the individual report options together, ensure that the same report option is not added twice.
- Check that conflicting report options are not specified. For example, do not add both MQRO_EXCEPTION and MQRO_EXCEPTION_WITH_DATA to the `Report` field; only one of these can be specified.

z/OS 2105 (0839) (RC2105): MQRC_STORAGE_CLASS_ERROR

Explanation

The MQPUT or MQPUT1 call was issued, but the storage-class object defined for the queue does not exist.

This reason code occurs only on z/OS.

Completion code

MQCC_FAILED

Programmer response

Create the storage-class object required by the queue, or modify the queue definition to use an existing storage class. The name of the storage-class object used by the queue is given by the **`StorageClass`** queue attribute.

z/OS 2106 (083A) (RC2106): MQRC_COD_NOT_VALID_FOR_XCF_Q

Explanation

An MQPUT or MQPUT1 call was issued, but the `Report` field in the message descriptor MQMD specifies one of the MQRO_COD_* options and the target queue is an XCF queue. MQRO_COD_* options cannot be specified for XCF queues.

This reason code occurs only on z/OS.

Completion code

MQCC_FAILED

Programmer response

Remove the relevant MQRO_COD_* option.

2107 (083B) (RC2107): MQRC_XWAIT_CANCELED

Explanation

An MQXWAIT call was issued, but the call has been canceled because a **STOP CHINIT** command has been issued (or the queue manager has been stopped, which causes the same effect). See MQXWAIT for more information about the MQXWAIT call.

The MQRC_XWAIT_CANCELED return code might also be used internally by IBM MQ on Distributed platforms.

Completion code

MQCC_FAILED

Programmer response

Tidy up and terminate.

z/OS 2108 (083C) (RC2108): MQRC_XWAIT_ERROR

Explanation

An MQXWAIT call was issued, but the invocation was not valid for one of the following reasons:

- The wait descriptor MQXWD contains data that is not valid.
- The linkage stack level is not valid.
- The addressing mode is not valid.
- There are too many wait events outstanding.

This reason code occurs only on z/OS.

Completion code

MQCC_FAILED

Programmer response

Obey the rules for using the MQXWAIT call. For more information about MQWAIT, see MQXWAIT.

2109 (083D) (RC2109): MQRC_SUPPRESSED_BY_EXIT

Explanation

On any call other than MQCONN or MQDISC, the API crossing exit suppressed the call.

Completion code

MQCC_FAILED

Programmer response

Obey the rules for MQI calls that the exit enforces. To find out the rules, see the writer of the exit.

2110 (083E) (RC2110): MQRC_FORMAT_ERROR

Explanation

An MQGET call was issued with the MQGMO_CONVERT option specified in the **GetMsgOpts** parameter, but the message cannot be converted successfully due to an error associated with the message format. Possible errors include:

- The format name in the message is MQFMT_NONE.
- A user-written exit with the name specified by the `Format` field in the message cannot be found.
- The message contains data that is not consistent with the format definition.

The message is returned unconverted to the application issuing the MQGET call, the values of the `CodedCharSetId` and `Encoding` fields in the **MsgDesc** parameter are set to those of the message returned, and the call completes with MQCC_WARNING.

If the message consists of several parts, each of which is described by its own `CodedCharSetId` and `Encoding` fields (for example, a message with format name MQFMT_DEAD_LETTER_HEADER), some parts may be converted and other parts not converted. However, the values returned in the various `CodedCharSetId` and `Encoding` fields always correctly describe the relevant message data.

Completion code

MQCC_WARNING

Programmer response

Check the format name that was specified when the message was put. If this is not one of the built-in formats, check that a suitable exit with the same name as the format is available for the queue manager to load. Verify that the data in the message corresponds to the format expected by the exit.

2111 (083F) (RC2111): MQRC_SOURCE_CCSID_ERROR

Explanation

The coded character-set identifier from which character data is to be converted is not valid or not supported.

This can occur on the MQGET call when the MQGMO_CONVERT option is included in the **GetMsgOpts** parameter; the coded character-set identifier in error is the `CodedCharSetId` field in the message being retrieved. In this case, the message data is returned unconverted, the values of the `CodedCharSetId` and `Encoding` fields in the **MsgDesc** parameter are set to those of the message returned, and the call completes with MQCC_WARNING.

This reason can also occur on the MQGET call when the message contains one or more MQ header structures (MQCIH, MQDLH, MQIIH, MQRMH), and the `CodedCharSetId` field in the message specifies a character set that does not have SBCS characters for the characters that are valid in queue names. MQ header structures containing such characters are not valid, and so the message is returned unconverted. The Unicode character set UTF-16 is an example of such a character set.

If the message consists of several parts, each of which is described by its own CodedCharSetId and Encoding fields (for example, a message with format name MQFMT_DEAD_LETTER_HEADER), some parts may be converted and other parts not converted. However, the values returned in the various CodedCharSetId and Encoding fields always correctly describe the relevant message data.

This reason can also occur on the MQXCNVC call; the coded character-set identifier in error is the **SourceCCSID** parameter. Either the **SourceCCSID** parameter specifies a value that is not valid or not supported, or the **SourceCCSID** parameter pointer is not valid. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)

This reason can also occur on a MQSETMP/MQINQMP/MQDLTMP call when the application issuing the calls does not use Language Environment (LE) and defines CCSID values of MQCCSI_APPL (-3) for message property names and string property values.

Completion code

MQCC_WARNING or MQCC_FAILED

Programmer response

Check the character-set identifier that was specified when the message was put, or that was specified for the **SourceCCSID** parameter on the MQXCNVC call. If this is correct, check that it is one for which queue manager conversion is supported. If queue manager conversion is not supported for the specified character set, conversion must be carried out by the application.

If this reason happens as result of a MQSETMP/MQINQMP/MQDLTMP call issued in a non-LE application program that has specified CCSID as MQCCSI_APPL (-3) then applications should be changed to specify the CCSID value used by the application to encode the property names or property string values.

Your applications should override the value of MQCCSI_APPL (-3) with the correct CCSID used as described in Redefinition of MQCCSI_APPL, or they should set the explicit CCSID value used to encode text strings in MQCHARV or similar structures.

2112 (0840) (RC2112): MQRC_SOURCE_INTEGER_ENC_ERROR

Explanation

On an MQGET call, with the MQGMO_CONVERT option included in the **GetMsgOpts** parameter, the Encoding value in the message being retrieved specifies an integer encoding that is not recognized. The message data is returned unconverted, the values of the CodedCharSetId and Encoding fields in the **MsgDesc** parameter are set to those of the message returned, and the call completes with MQCC_WARNING.

If the message consists of several parts, each of which is described by its own CodedCharSetId and Encoding fields (for example, a message with format name MQFMT_DEAD_LETTER_HEADER), some parts may be converted and other parts not converted. However, the values returned in the various CodedCharSetId and Encoding fields always correctly describe the relevant message data.

This reason code can also occur on the MQXCNVC call, when the **Options** parameter contains an unsupported MQDCC_SOURCE_* value, or when MQDCC_SOURCE_ENC_UNDEFINED is specified for a UTF-16 code page.

Completion code

MQCC_WARNING or MQCC_FAILED

Programmer response

Check the integer encoding that was specified when the message was put. If this is correct, check that it is one for which queue manager conversion is supported. If queue manager conversion is not supported for the required integer encoding, conversion must be carried out by the application.

2113 (0841) (RC2113): MQRC_SOURCE_DECIMAL_ENC_ERROR

Explanation

On an MQGET call with the MQGMO_CONVERT option included in the **GetMsgOpts** parameter, the *Encoding* value in the message being retrieved specifies a decimal encoding that is not recognized. The message data is returned unconverted, the values of the CodedCharSetId and Encoding fields in the **MsgDesc** parameter are set to those of the message returned, and the call completes with MQCC_WARNING.

If the message consists of several parts, each of which is described by its own CodedCharSetId and Encoding fields (for example, a message with format name MQFMT_DEAD_LETTER_HEADER), some parts may be converted and other parts not converted. However, the values returned in the various CodedCharSetId and Encoding fields always correctly describe the relevant message data.

Completion code

MQCC_WARNING

Programmer response

Check the decimal encoding that was specified when the message was put. If this is correct, check that it is one for which queue manager conversion is supported. If queue manager conversion is not supported for the required decimal encoding, conversion must be carried out by the application.

2114 (0842) (RC2114): MQRC_SOURCE_FLOAT_ENC_ERROR

Explanation

On an MQGET call, with the MQGMO_CONVERT option included in the **GetMsgOpts** parameter, the Encoding value in the message being retrieved specifies a floating-point encoding that is not recognized. The message data is returned unconverted, the values of the CodedCharSetId and Encoding fields in the **MsgDesc** parameter are set to those of the message returned, and the call completes with MQCC_WARNING.

If the message consists of several parts, each of which is described by its own CodedCharSetId and Encoding fields (for example, a message with format name MQFMT_DEAD_LETTER_HEADER), some parts may be converted and other parts not converted. However, the values returned in the various CodedCharSetId and Encoding fields always correctly describe the relevant message data.

Completion code

MQCC_WARNING

Programmer response

Check the floating-point encoding that was specified when the message was put. If this is correct, check that it is one for which queue manager conversion is supported. If queue manager conversion is not supported for the required floating-point encoding, conversion must be carried out by the application.

2115 (0843) (RC2115): MQRC_TARGET_CCSID_ERROR

Explanation

The coded character-set identifier to which character data is to be converted is not valid or not supported.

This can occur on the MQGET call when the MQGMO_CONVERT option is included in the **GetMsgOpts** parameter; the coded character-set identifier in error is the CodedCharSetId field in the **MsgDesc** parameter. In this case, the message data is returned unconverted, the values of the CodedCharSetId and Encoding fields in the **MsgDesc** parameter are set to those of the message returned, and the call completes with MQCC_WARNING.

This reason can also occur on the MQGET call when the message contains one or more MQ header structures (MQCIH, MQDLH, MQIIH, MQRMH), and the CodedCharSetId field in the **MsgDesc** parameter specifies a character set that does not have SBCS characters for the characters that are valid in queue names. The Unicode character set UTF-16 is an example of such a character set.

This reason can also occur on the MQXCNVC call; the coded character-set identifier in error is the **TargetCCSID** parameter. Either the **TargetCCSID** parameter specifies a value that is not valid or not supported, or the **TargetCCSID** parameter pointer is not valid. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)

Completion code

MQCC_WARNING or MQCC_FAILED

Programmer response

Check the character-set identifier that was specified for the CodedCharSetId field in the **MsgDesc** parameter on the MQGET call, or that was specified for the **SourceCCSID** parameter on the MQXCNVC call. If this is correct, check that it is one for which queue manager conversion is supported. If queue manager conversion is not supported for the specified character set, conversion must be carried out by the application.

2116 (0844) (RC2116): MQRC_TARGET_INTEGER_ENC_ERROR

Explanation

On an MQGET call with the MQGMO_CONVERT option included in the **GetMsgOpts** parameter, the Encoding value in the **MsgDesc** parameter specifies an integer encoding that is not recognized. The message data is returned unconverted, the values of the CodedCharSetId and Encoding fields in the **MsgDesc** parameter are set to those of the message being retrieved, and the call completes with MQCC_WARNING.

This reason code can also occur on the MQXCNVC call, when the **Options** parameter contains an unsupported MQDCC_TARGET_* value, or when MQDCC_TARGET_ENC_UNDEFINED is specified for a UTF-16 code page.

Completion code

MQCC_WARNING or MQCC_FAILED

Programmer response

Check the integer encoding that was specified. If this is correct, check that it is one for which queue manager conversion is supported. If queue manager conversion is not supported for the required integer encoding, conversion must be carried out by the application.

2117 (0845) (RC2117): MQRC_TARGET_DECIMAL_ENC_ERROR

Explanation

On an MQGET call with the MQGMO_CONVERT option included in the **GetMsgOpts** parameter, the Encoding value in the **MsgDesc** parameter specifies a decimal encoding that is not recognized. The message data is returned unconverted, the values of the CodedCharSetId and Encoding fields in the **MsgDesc** parameter are set to those of the message returned, and the call completes with MQCC_WARNING.

Completion code

MQCC_WARNING

Programmer response

Check the decimal encoding that was specified. If this is correct, check that it is one for which queue manager conversion is supported. If queue manager conversion is not supported for the required decimal encoding, conversion must be carried out by the application.

2118 (0846) (RC2118): MQRC_TARGET_FLOAT_ENC_ERROR

Explanation

On an MQGET call with the MQGMO_CONVERT option included in the **GetMsgOpts** parameter, the Encoding value in the **MsgDesc** parameter specifies a floating-point encoding that is not recognized. The message data is returned unconverted, the values of the CodedCharSetId and Encoding fields in the **MsgDesc** parameter are set to those of the message returned, and the call completes with MQCC_WARNING.

Completion code

MQCC_WARNING

Programmer response

Check the floating-point encoding that was specified. If this is correct, check that it is one for which queue manager conversion is supported. If queue manager conversion is not supported for the required floating-point encoding, conversion must be carried out by the application.

2119 (0847) (RC2119): MQRC_NOT_CONVERTED

Explanation

An MQGET call was issued with the MQGMO_CONVERT option specified in the **GetMsgOpts** parameter, but an error occurred during conversion of the data in the message. The message data is returned unconverted, the values of the CodedCharSetId and Encoding fields in the **MsgDesc** parameter are set to those of the message returned, and the call completes with MQCC_WARNING.

If the message consists of several parts, each of which is described by its own CodedCharSetId and Encoding fields (for example, a message with format name MQFMT_DEAD_LETTER_HEADER), some parts may be converted and other parts not converted. However, the values returned in the various CodedCharSetId and Encoding fields always correctly describe the relevant message data.

This error may also indicate that a parameter to the data-conversion service is not supported.

Completion code

MQCC_WARNING

Programmer response

Check that the message data is correctly described by the **Format**, **CodedCharSetId** and **Encoding** parameters that were specified when the message was put. Also check that these values, and the `CodedCharSetId` and `Encoding` specified in the **MsgDesc** parameter on the MQGET call, are supported for queue manager conversion. If the required conversion is not supported, conversion must be carried out by the application.

2120 (0848) (RC2120): MQRC_CONVERTED_MSG_TOO_BIG

Explanation

On an MQGET call with the MQGMO_CONVERT option included in the **GetMsgOpts** parameter, the message data expanded during data conversion and exceeded the size of the buffer provided by the application. However, the message had already been removed from the queue because prior to conversion the message data could be accommodated in the application buffer without truncation.

The message is returned unconverted, with the **CompCode** parameter of the MQGET call set to MQCC_WARNING. If the message consists of several parts, each of which is described by its own character-set and encoding fields (for example, a message with format name MQFMT_DEAD_LETTER_HEADER), some parts may be converted and other parts not converted. However, the values returned in the various character-set and encoding fields always correctly describe the relevant message data.

This reason also occurs on the MQXCNVC call, when the **TargetBuffer** parameter is too small to accommodate the converted string, and the string has been truncated to fit in the buffer. The length of valid data returned is given by the **DataLength** parameter; in the case of a DBCS string or mixed SBCS/DBCS string, this length may be less than the length of **TargetBuffer**.

Completion code

MQCC_WARNING

Programmer response

For the MQGET call, check that the exit is converting the message data correctly and setting the output length `DataLength` to the appropriate value. If it is, the application issuing the MQGET call must provide a larger buffer for the **Buffer** parameter.

For the MQXCNVC call, if the string must be converted without truncation, provide a larger output buffer.

Multi 2121 (0849) (RC2121): MQRC_NO_EXTERNAL_PARTICIPANTS

Explanation

Multi

An MQBEGIN call was issued to start a unit of work coordinated by the queue manager, but no participating resource managers have been registered with the queue manager. As a result, only changes to IBM MQ resources can be coordinated by the queue manager in the unit of work.

Completion code

MQCC_WARNING

Programmer response

If the application does not require non-MQ resources to participate in the unit of work, this reason code can be ignored or the MQBEGIN call removed. Otherwise consult your system programmer to determine why the required resource managers have not been registered with the queue manager; the queue manager's configuration file might be in error.

Multi 2122 (084A) (RC2122): MQRC_PARTICIPANT_NOT_AVAILABLE

Explanation

Multi

An MQBEGIN call was issued to start a unit of work coordinated by the queue manager, but one or more of the participating resource managers that had been registered with the queue manager is not available. As a result, changes to those resources cannot be coordinated by the queue manager in the unit of work.

Completion code

MQCC_WARNING

Programmer response

If the application does not require non-MQ resources to participate in the unit of work, this reason code can be ignored. Otherwise consult your system programmer to determine why the required resource managers are not available. The resource manager might have been halted temporarily, or there might be an error in the queue manager's configuration file.

ALW 2123 (084B) (RC2123): MQRC_OUTCOME_MIXED

Explanation

The queue manager is acting as the unit-of-work coordinator for a unit of work that involves other resource managers, but one of the following occurred:

- An MQCMIT or MQDISC call was issued to commit the unit of work, but one or more of the participating resource managers backed-out the unit of work instead of committing it. As a result, the outcome of the unit of work is mixed.
- An MQBACK call was issued to back out a unit of work, but one or more of the participating resource managers had already committed the unit of work.

Completion code

MQCC_FAILED

Programmer response

Examine the queue manager error logs for messages relating to the mixed outcome; these messages identify the resource managers that are affected. Use procedures local to the affected resource managers to resynchronize the resources.

This reason code does not prevent the application initiating further units of work.

z/OS ALW 2124 (084C) (RC2124): MQRC_OUTCOME_PENDING

Explanation

z/OS ALW

The queue manager is acting as the unit-of-work coordinator for a unit of work that involves other resource managers, and an MQCMIT or MQDISC call was issued to commit the unit of work, but one or more of the participating resource managers has not confirmed that the unit of work was committed successfully.

The completion of the commit operation will happen at some point in the future, but there remains the possibility that the outcome will be mixed.

z/OS On z/OS, this situation can occur if a queue manager loses connectivity to a coupling facility structure while a unit of work that affects messages on shared queues is being committed or backed out.

Completion code

MQCC_WARNING

Programmer response

Windows AIX Use the normal error-reporting mechanisms to determine whether the outcome was mixed. If it was, take appropriate action to resynchronize the resources.

Windows AIX This reason code does not prevent the application initiating further units of work.

z/OS If this reason code was returned as a result of loss of connectivity to a coupling facility structure on z/OS, the operation will be completed either when the queue manager reconnects to the affected structure, or when another queue manager in the queue sharing group is able to perform peer recovery on the structure.

2125 (084D) (RC2125): MQRC_BRIDGE_STARTED

Explanation

z/OS

The IMS bridge has been started.

Completion code

MQCC_WARNING

Programmer response

None. This reason code is only used to identify the corresponding event message “Bridge Started” on page 141.

2126 (084E) (RC2126): MQRC_BRIDGE_STOPPED

Explanation

z/OS

The IMS bridge has been stopped.

Completion code

MQCC_WARNING

Programmer response

None. This reason code is only used to identify the corresponding event message “Bridge Stopped” on page 142.

z/OS 2127 (084F) (RC2127): MQRC_ADAPTER_STORAGE_SHORTAGE

Explanation

On an MQCONN call, the adapter was unable to acquire storage.

This reason code occurs only on z/OS.

Completion code

MQCC_FAILED

Programmer response

Notify the system programmer. The system programmer should determine why the system is short on storage, and take appropriate action, for example, increase the region size on the step or job card.

Multi 2128 (0850) (RC2128): MQRC_UOW_IN_PROGRESS

Explanation

Multi

An MQBEGIN call was issued to start a unit of work coordinated by the queue manager, but a unit of work is already in existence for the connection handle specified. This may be a global unit of work started by a previous MQBEGIN call, or a unit of work that is local to the queue manager or one of the cooperating resource managers. No more than one unit of work can exist concurrently for a connection handle.

Completion code

MQCC_FAILED

Programmer response

Review the application logic to determine why there is a unit of work already in existence. Move the MQBEGIN call to the appropriate place in the application.

z/OS 2129 (0851) (RC2129): MQRC_ADAPTER_CONN_LOAD_ERROR

Explanation

On an MQCONN call, the connection handling module could not be loaded, so the adapter could not link to it. The connection handling module name is:

- CSQBCON for batch applications

- CSQQCONN or CSQQCON2 for IMS applications

This reason code occurs only on z/OS.

Completion code

MQCC_FAILED

Programmer response

Ensure that the correct library concatenation has been specified in the batch application program execution JCL, and in the queue manager startup JCL.

z/OS 2130 (0852) (RC2130): MQRC_ADAPTER_SERV_LOAD_ERROR

Explanation

On an MQI call, the batch adapter could not load one of the following API service module, and so could not link to it:

- CSQBSRV
- CSQAPEPL
- CSQBCRMH
- CSQBAPPL

This reason code occurs only on z/OS.

Completion code

MQCC_FAILED

Programmer response

Ensure that the correct library concatenation has been specified in the batch application program execution JCL, and in the queue manager startup JCL.

z/OS 2131 (0853) (RC2131): MQRC_ADAPTER_DEFS_ERROR

Explanation

On an MQCONN call, the subsystem definition module (CSQBDEFV for batch and CSQQDEFV for IMS) does not contain the required control block identifier.

This reason code occurs only on z/OS.

Completion code

MQCC_FAILED

Programmer response

Check your library concatenation. If this is correct, check that the CSQBDEFV or CSQQDEFV module contains the required subsystem ID.

z/OS 2132 (0854) (RC2132): MQRC_ADAPTER_DEFS_LOAD_ERROR

Explanation

On an MQCONN call, the subsystem definition module (CSQBDEFV for batch and CSQQDEFV for IMS) could not be loaded.

This reason code occurs only on z/OS.

Completion code

MQCC_FAILED

Programmer response

Ensure that the correct library concatenation has been specified in the application program execution JCL, and in the queue manager startup JCL.

z/OS 2133 (0855) (RC2133): MQRC_ADAPTER_CONV_LOAD_ERROR

Explanation

On an MQGET call, the adapter (batch or IMS) could not load the data conversion services modules.

This reason code occurs only on z/OS.

Completion code

MQCC_FAILED

Programmer response

Ensure that the correct library concatenation has been specified in the batch application program execution JCL, and in the queue manager startup JCL.

Multi 2134 (0856) (RC2134): MQRC_BO_ERROR

Explanation

On an MQBEGIN call, the begin-options structure MQBO is not valid, for one of the following reasons:

- The `StrucId` field is not MQBO_STRUC_ID.
- The `Version` field is not MQBO_VERSION_1.
- The parameter pointer is not valid. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)
- The queue manager cannot copy the changed structure to application storage, even though the call is successful. This can occur, for example, if the pointer points to read-only storage.

Completion code

MQCC_FAILED

Programmer response

Ensure that input fields in the MQBO structure are set correctly.

Multi 2135 (0857) (RC2135): MQRC_DH_ERROR

Explanation

An MQPUT or MQPUT1 call was issued, but the message data contains an MQDH structure that is not valid. Possible errors include the following:

- The `StrucId` field is not MQDH_STRUC_ID.
- The `Version` field is not MQDH_VERSION_1.
- The `StrucLength` field specifies a value that is too small to include the structure plus the arrays of MQOR and MQPMR records.
- The `CodedCharSetId` field is zero, or a negative value that is not valid.
- The **`BufferLength`** parameter of the call has a value that is too small to accommodate the structure (the structure extends beyond the end of the message).

Completion code

MQCC_FAILED

Programmer response

Check that the fields in the structure are set correctly. Ensure that the application sets the `CodedCharSetId` field to a valid value (note: MQCCSI_DEFAULT, MQCCSI_EMBEDDED, MQCCSI_Q_MGR, and MQCCSI_UNDEFINED are not valid in this field).

Multi 2136 (0858) (RC2136): MQRC_MULTIPLE_REASONS

Explanation

An MQOPEN, MQPUT or MQPUT1 call was issued to open a distribution list or put a message to a distribution list, but the result of the call was not the same for all of the destinations in the list. One of the following applies:

- The call succeeded for some of the destinations but not others. The completion code is MQCC_WARNING in this case.
- The call failed for all of the destinations, but for differing reasons. The completion code is MQCC_FAILED in this case.

Completion code

MQCC_WARNING or MQCC_FAILED

Programmer response

Examine the MQRR response records to identify the destinations for which the call failed, and the reason for the failure. Ensure that sufficient response records are provided by the application on the call to enable the error(s) to be determined. For the MQPUT1 call, the response records must be specified using the MQOD structure, and not the MQPMO structure.

Multi 2137 (0859) (RC2137): MQRC_OPEN_FAILED

Explanation

A queue or other MQ object could not be opened successfully, for one of the following reasons:

- An MQCONN or MQCONNX call was issued, but the queue manager was unable to open an object that is used internally by the queue manager. As a result, processing cannot continue. The error log will contain the name of the object that could not be opened.
- An MQPUT call was issued to put a message to a distribution list, but the message could not be sent to the destination to which this reason code applies because that destination was not opened successfully by the MQOPEN call. This reason occurs only in the *Reason* field of the MQRR response record.

Completion code

MQCC_FAILED

Programmer response

Do one of the following:

- If the error occurred on the MQCONN or MQCONNX call, ensure that the required objects exist by running the following command and then retrying the application:

  ```
  STRMQM -c qmgr
  ```

  where `qmgr` should be replaced by the name of the queue manager.
- If the error occurred on the MQPUT call, examine the MQRR response records specified on the MQOPEN call to determine the reason that the queue failed to open. Ensure that sufficient response records are provided by the application on the call to enable the error(s) to be determined.

z/OS 2138 (085A) (RC2138): MQRC_ADAPTER_DISC_LOAD_ERROR

Explanation

On an MQDISC call, the disconnect handling module (CSQBDSC for batch and CSQQDISC for IMS) could not be loaded, so the adapter could not link to it.

This reason code occurs only on z/OS.

Completion code

MQCC_FAILED

Programmer response

Ensure that the correct library concatenation has been specified in the application program execution JCL, and in the queue manager startup JCL. Any uncommitted changes in a unit of work should be backed out. A unit of work that is coordinated by the queue manager is backed out automatically.

Multi 2139 (085B) (RC2139): MQRC_CNO_ERROR

Explanation

On an MQCONNX call, the connect-options structure MQCNO is not valid, for one of the following reasons:

- The *StrucId* field is not MQCNO_STRUC_ID.
- The *Version* field specifies a value that is not valid or not supported.
- The parameter pointer is not valid. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)
- The queue manager cannot copy the changed structure to application storage, even though the call is successful. This can occur, for example, if the parameter pointer points to read-only storage.

Completion code

MQCC_FAILED

Programmer response

Ensure that input fields in the MQCNO structure are set correctly.

z/OS 2140 (085C) (RC2140): MQRC_CICS_WAIT_FAILED

Explanation

On any MQI call, the CICS adapter issued an EXEC CICS WAIT request, but the request was rejected by CICS.

This reason code occurs only on z/OS.

Completion code

MQCC_FAILED

Programmer response

Examine the CICS trace data for actual response codes. The most likely cause is that the task has been canceled by the operator or by the system.

2141 (085D) (RC2141): MQRC_DLH_ERROR

Explanation

An MQPUT or MQPUT1 call was issued, but the message data contains an MQDLH structure that is not valid. Possible errors include the following:

- The `StrucId` field is not MQDLH_STRUC_ID.
- The `Version` field is not MQDLH_VERSION_1.
- The `CodedCharSetId` field is zero, or a negative value that is not valid.
- The **`BufferLength`** parameter of the call has a value that is too small to accommodate the structure (the structure extends beyond the end of the message).

Completion code

MQCC_FAILED

Programmer response

Check that the fields in the structure are set correctly. Ensure that the application sets the `CodedCharSetId` field to a valid value (note: MQCCSI_DEFAULT, MQCCSI_EMBEDDED, MQCCSI_Q_MGR, and MQCCSI_UNDEFINED are not valid in this field).

2142 (085E) (RC2142): MQRC_HEADER_ERROR

Explanation

An MQPUT or MQPUT1 call was issued, but the message data contains an MQ header structure that is not valid. Possible errors include the following:

- The `StrucId` field is not valid.
- The `Version` field is not valid.
- The `StrucLength` field specifies a value that is too small.
- The `CodedCharSetId` field is zero, or a negative value that is not valid.
- The **BufferLength** parameter of the call has a value that is too small to accommodate the structure (the structure extends beyond the end of the message).

Completion code

MQCC_FAILED

Programmer response

Check that the fields in the structure are set correctly. Ensure that the application sets the *CodedCharSetId* field to a valid value (note: MQCCSI_DEFAULT, MQCCSI_EMBEDDED, MQCCSI_Q_MGR, and MQCCSI_UNDEFINED are not valid in this field).

2143 (085F) (RC2143): MQRC_SOURCE_LENGTH_ERROR

Explanation

On the MQXCNVC call, the **SourceLength** parameter specifies a length that is less than zero or not consistent with the string's character set or content (for example, the character set is a double-byte character set, but the length is not a multiple of two). This reason also occurs if the **SourceLength** parameter pointer is not valid. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)

This reason code can also occur on the MQGET call when the MQGMO_CONVERT option is specified. In this case it indicates that the MQRC_SOURCE_LENGTH_ERROR reason was returned by an MQXCNVC call issued by the data conversion exit.

Completion code

MQCC_WARNING or MQCC_FAILED

Programmer response

Specify a length that is zero or greater. If the reason code occurs on the MQGET call, check that the logic in the data-conversion exit is correct.

2144 (0860) (RC2144): MQRC_TARGET_LENGTH_ERROR

Explanation

On the MQXCNVC call, the **TargetLength** parameter is not valid for one of the following reasons:

- **TargetLength** is less than zero.
- The **TargetLength** parameter pointer is not valid. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)
- The MQDCC_FILL_TARGET_BUFFER option is specified, but the value of **TargetLength** is such that the target buffer cannot be filled completely with valid characters. This can occur when **TargetCCSID** is a pure DBCS character set (such as UTF-16), but **TargetLength** specifies a length that is an odd number of bytes.

This reason code can also occur on the MQGET call when the MQGMO_CONVERT option is specified. In this case it indicates that the MQRC_TARGET_LENGTH_ERROR reason was returned by an MQXCNVC call issued by the data conversion exit.

Completion code

MQCC_WARNING or MQCC_FAILED

Programmer response

Specify a length that is zero or greater. If the MQDCC_FILL_TARGET_BUFFER option is specified, and **TargetCCSID** is a pure DBCS character set, ensure that **TargetLength** specifies a length that is a multiple of two.

If the reason code occurs on the MQGET call, check that the logic in the data-conversion exit is correct.

2145 (0861) (RC2145): MQRC_SOURCE_BUFFER_ERROR

Explanation

On the MQXCNVC call, the **SourceBuffer** parameter pointer is not valid, or points to storage that cannot be accessed for the entire length specified by **SourceLength**. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)

This reason code can also occur on the MQGET call when the MQGMO_CONVERT option is specified. In this case it indicates that the MQRC_SOURCE_BUFFER_ERROR reason was returned by an MQXCNVC call issued by the data conversion exit.

Completion code

MQCC_WARNING or MQCC_FAILED

Programmer response

Specify a valid buffer. If the reason code occurs on the MQGET call, check that the logic in the data-conversion exit is correct.

2146 (0862) (RC2146): MQRC_TARGET_BUFFER_ERROR

Explanation

On the MQXCNVC call, the **TargetBuffer** parameter pointer is not valid, or points to read-only storage, or to storage that cannot be accessed for the entire length specified by **TargetLength**. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)

This reason code can also occur on the MQGET call when the MQGMO_CONVERT option is specified. In this case it indicates that the MQRC_TARGET_BUFFER_ERROR reason was returned by an MQXCNVC call issued by the data conversion exit.

Completion code

MQCC_WARNING or MQCC_FAILED

Programmer response

Specify a valid buffer. If the reason code occurs on the MQGET call, check that the logic in the data-conversion exit is correct.

Windows 2147 (0863) (RC2147): MQRC_INCOMPLETE_TRANSACTION

Explanation

An attempt was made to disconnect a connection handle that is still participating in a DTC transaction.

This can occur when a managed IBM MQ .NET application tries to disconnect the connection handle before completing the transaction.

This error does not occur for non-transactional MQI calls.

This reason code occurs only on Windows.

Completion code

MQCC_FAILED

Programmer response

Check the application design and ensure the transaction that is associated with a queue manager connection is committed or rolled back before it is disconnected.

2148 (0864) (RC2148): MQRC_IIH_ERROR

Explanation

An MQPUT or MQPUT1 call was issued, but the message data contains an MQIIH structure that is not valid. Possible errors include the following:

- The `StrucId` field is not MQIIH_STRUC_ID.
- The `Version` field is not MQIIH_VERSION_1.
- The `StrucLength` field is not MQIIH_LENGTH_1.
- The **`BufferLength`** parameter of the call has a value that is too small to accommodate the structure (the structure extends beyond the end of the message).

Completion code

MQCC_FAILED

Programmer response

Check that the fields in the structure are set correctly.

2149 (0865) (RC2149): MQRC_PCF_ERROR

Explanation

An MQPUT or MQPUT1 call was issued to put a message containing PCF data, but the length of the message does not equal the sum of the lengths of the PCF structures present in the message. This can occur for messages with the following format names:

- MQFMT_ADMIN
- MQFMT_EVENT
- MQFMT_PCF

Completion code

MQCC_FAILED

Programmer response

Ensure that the length of the message specified on the MQPUT or MQPUT1 call equals the sum of the lengths of the PCF structures contained within the message data.

2150 (0866) (RC2150): MQRC_DBCS_ERROR

Explanation

An error was encountered attempting to convert a double-byte character set (DBCS) or variable-width encoding character set string. This can occur in the following cases:

- On the MQXCNVC call, when the **SourceCCSID** parameter specifies the coded character-set identifier of a double-byte or variable-width encoding character set, but the **SourceBuffer** parameter does not contain a valid string. This might be because the string contains characters that are not valid, or because the string is a mixed SBCS/DBCS string and the shift-out/shift-in characters are not correctly paired. The completion code is MQCC_FAILED in this case.
- On the MQGET call, when the MQGMO_CONVERT option is specified. In this case it indicates that the MQRC_DBCS_ERROR reason code was returned by an MQXCNVC call issued by the data conversion exit. The completion code is MQCC_WARNING in this case.
- z/OS For the z/OS dead letter handler utility CSQUDLQH, when the rule being processed uses the default of CONVERT(YES). Modify the rule to use CONVERT(NO) if the data does not need to be converted.

Completion code

MQCC_WARNING or MQCC_FAILED

Programmer response

Specify a valid string.

If the reason code occurs on the MQGET call, check that the data in the message is valid, and that the logic in the data-conversion exit is correct.

Multi 2152 (0868) (RC2152): MQRC_OBJECT_NAME_ERROR

Explanation

An MQOPEN or MQPUT1 call was issued to open a distribution list (that is, the `RecsPresent` field in MQOD is greater than zero), but the `ObjectName` field is neither blank nor the null string.

Completion code

MQCC_FAILED

Programmer response

If it is intended to open a distribution list, set the `ObjectName` field to blanks or the null string. If it is not intended to open a distribution list, set the `RecsPresent` field to zero.

Multi 2153 (0869) (RC2153): MQRC_OBJECT_Q_MGR_NAME_ERROR

Explanation

An MQOPEN or MQPUT1 call was issued to open a distribution list (that is, the `RecsPresent` field in MQOD is greater than zero), but the `ObjectQMgrName` field is neither blank nor the null string.

Completion code

MQCC_FAILED

Programmer response

If it is intended to open a distribution list, set the `ObjectQMgrName` field to blanks or the null string. If it is not intended to open a distribution list, set the `RecsPresent` field to zero.

Multi 2154 (086A) (RC2154): MQRC_RECS_PRESENT_ERROR

Explanation

An MQOPEN or MQPUT1 call was issued, but the call failed for one of the following reasons:

- `RecsPresent` in MQOD is less than zero.
- `ObjectType` in MQOD is not MQOT_Q, and `RecsPresent` is not zero. `RecsPresent` must be zero if the object being opened is not a queue.
- IBM MQ Multicast is being used and `RecsPresent` in MQOD is not set to zero. IBM MQ Multicast does not use distribution lists.

Completion code

MQCC_FAILED

Programmer response

If it is intended to open a distribution list, set the `ObjectType` field to MQOT_Q and `RecsPresent` to the number of destinations in the list. If it is not intended to open a distribution list, set the `RecsPresent` field to zero.

Multi 2155 (086B) (RC2155): MQRC_OBJECT_RECORDS_ERROR

Explanation

An MQOPEN or MQPUT1 call was issued to open a distribution list (that is, the `RecsPresent` field in MQOD is greater than zero), but the MQOR object records are not specified correctly. One of the following applies:

- `ObjectRecOffset` is zero and `ObjectRecPtr` is zero or the null pointer.
- `ObjectRecOffset` is not zero and `ObjectRecPtr` is not zero and not the null pointer.
- `ObjectRecPtr` is not a valid pointer.
- `ObjectRecPtr` or `ObjectRecOffset` points to storage that is not accessible.

Completion code

MQCC_FAILED

Programmer response

Ensure that one of `ObjectRecOffset` and `ObjectRecPtr` is zero and the other nonzero. Ensure that the field used points to accessible storage.

Multi 2156 (086C) (RC2156): MQRC_RESPONSE_RECORDS_ERROR

Explanation

An MQOPEN or MQPUT1 call was issued to open a distribution list (that is, the `RecsPresent` field in MQOD is greater than zero), but the MQRR response records are not specified correctly. One of the following applies:

- `ResponseRecOffset` is not zero and `ResponseRecPtr` is not zero and not the null pointer.
- `ResponseRecPtr` is not a valid pointer.
- `ResponseRecPtr` or `ResponseRecOffset` points to storage that is not accessible.

Completion code

MQCC_FAILED

Programmer response

Ensure that at least one of `ResponseRecOffset` and `ResponseRecPtr` is zero. Ensure that the field used points to accessible storage.

z/OS 2157 (086D) (RC2157): MQRC_ASID_MISMATCH

Explanation

On any MQI call, the caller's primary ASID was found to be different from the home ASID.

This reason code occurs only on z/OS.

Completion code

MQCC_FAILED

Programmer response

Correct the application (MQI calls cannot be issued in cross-memory mode). Any uncommitted changes in a unit of work should be backed out. A unit of work that is coordinated by the queue manager is backed out automatically.

Multi 2158 (086E) (RC2158): MQRC_PMO_RECORD_FLAGS_ERROR

Explanation

An MQPUT or MQPUT1 call was issued to put a message, but the *PutMsgRecFields* field in the MQPMO structure is not valid, for one of the following reasons:

- The field contains flags that are not valid.
- The message is being put to a distribution list, and put message records have been provided (that is, *RecsPresent* is greater than zero, and one of *PutMsgRecOffset* or *PutMsgRecPtr* is nonzero), but *PutMsgRecFields* has the value MQPMRF_NONE.

- MQPMRF_ACCOUNTING_TOKEN is specified without either MQPMO_SET_IDENTITY_CONTEXT or MQPMO_SET_ALL_CONTEXT.

Completion code

MQCC_FAILED

Programmer response

Ensure that *PutMsgRecFields* is set with the appropriate MQPMRF_* flags to indicate which fields are present in the put message records. If MQPMRF_ACCOUNTING_TOKEN is specified, ensure that either MQPMO_SET_IDENTITY_CONTEXT or MQPMO_SET_ALL_CONTEXT is also specified. Alternatively, set both *PutMsgRecOffset* and *PutMsgRecPtr* to zero.

Multi 2159 (086F) (RC2159): MQRC_PUT_MSG_RECORDS_ERROR

Explanation

An MQPUT or MQPUT1 call was issued to put a message to a distribution list, but the MQPMR put message records are not specified correctly. One of the following applies:

- PutMsgRecOffset is not zero and PutMsgRecPtr is not zero and not the null pointer.
- PutMsgRecPtr is not a valid pointer.
- PutMsgRecPtr or PutMsgRecOffset points to storage that is not accessible.

Completion code

MQCC_FAILED

Programmer response

Ensure that at least one of PutMsgRecOffset and PutMsgRecPtr is zero. Ensure that the field used points to accessible storage.

z/OS 2160 (0870) (RC2160): MQRC_CONN_ID_IN_USE

Explanation

On an MQCONN call, the connection identifier assigned by the queue manager to the connection between a CICS or IMS allied address space and the queue manager conflicts with the connection identifier of another connected CICS or IMS system. The connection identifier assigned is as follows:

- For CICS, the applid
- For IMS, the IMSID parameter on the IMSCTRL (sysgen) macro, or the IMSID parameter on the execution parameter (EXEC card in IMS control region JCL)
- For batch, the job name
- For TSO, the user ID

A conflict arises only if there are two CICS systems, two IMS systems, or one each of CICS and IMS, having the same connection identifiers. Batch and TSO connections need not have unique identifiers.

This reason code occurs only on z/OS.

Completion code

MQCC_FAILED

Programmer response

Ensure that the naming conventions used in different systems that might connect to the queue manager do not conflict.

2161 (0871) (RC2161): MQRC_Q_MGR_QUIESCING

Explanation

An MQI call was issued, but the call failed because the queue manager is quiescing (preparing to shut down).

When the queue manager is quiescing, the MQOPEN, MQPUT, MQPUT1, and MQGET calls can still complete successfully, but the application can request that they fail by specifying the appropriate option on the call:

- MQOO_FAIL_IF_QUIESCING on MQOPEN
- MQPMO_FAIL_IF_QUIESCING on MQPUT or MQPUT1
- MQGMO_FAIL_IF_QUIESCING on MQGET

Specifying these options enables the application to become aware that the queue manager is preparing to shut down.

- z/OS On z/OS:
  - For batch applications, this reason can be returned to applications running in LPARs that do not have a queue manager installed.
  - For CICS applications, this reason can be returned when no connection was established.
- IBM i On IBM i, for applications running in compatibility mode, this reason can be returned when no connection was established.

Completion code

MQCC_FAILED

Programmer response

The application should tidy up and end. If the application specified the MQOO_FAIL_IF_QUIESCING, MQPMO_FAIL_IF_QUIESCING, or MQGMO_FAIL_IF_QUIESCING option on the failing call, the relevant option can be removed and the call reissued. By omitting these options, the application can continue working to complete and commit the current unit of work, but the application does not start a new unit of work.

2162 (0872) (RC2162): MQRC_Q_MGR_STOPPING

Explanation

An MQI call was issued, but the call failed because the queue manager is shutting down. If the call was an MQGET call with the MQGMO_WAIT option, the wait has been canceled. No more MQI calls can be issued.

For MQ MQI client applications, it is possible that the call did complete successfully, even though this reason code is returned with a *CompCode* of MQCC_FAILED.

z/OS On z/OS, the MQRC_CONNECTION_BROKEN reason may be returned instead if, as a result of system scheduling factors, the queue manager shuts down before the call completes.

Completion code

MQCC_FAILED

Programmer response

The application should tidy up and end. If the application is in the middle of a unit of work coordinated by an external unit-of-work coordinator, the application should issue the appropriate call to back out the unit of work. Any unit of work that is coordinated by the queue manager is backed out automatically.

z/OS 2163 (0873) (RC2163): MQRC_DUPLICATE_RECOV_COORD

Explanation

On an MQCONN or MQCONNX call, a recovery coordinator already exists for the connection name specified on the connection call issued by the adapter.

A conflict arises only if there are two CICS systems, two IMS systems, or one each of CICS and IMS, having the same connection identifiers. Batch and TSO connections need not have unique identifiers.

This reason code occurs only on z/OS.

Completion code

MQCC_FAILED

Programmer response

Ensure that the naming conventions used in different systems that might connect to the queue manager do not conflict.

2173 (087D) (RC2173): MQRC_PMO_ERROR

Explanation

On an MQPUT or MQPUT1 call, the MQPMO structure is not valid, for one of the following reasons:

- The `StrucId` field is not MQPMO_STRUC_ID.
- The `Version` field specifies a value that is not valid or not supported.
- The parameter pointer is not valid. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)
- The queue manager cannot copy the changed structure to application storage, even though the call is successful. This can occur, for example, if the pointer points to read-only storage.

Completion code

MQCC_FAILED

Programmer response

Ensure that input fields in the MQPMO structure are set correctly.

2182 (0886) (RC2182): MQRC_API_EXIT_NOT_FOUND

Explanation

The API crossing exit entry point could not be found.

Completion code

MQCC_FAILED

Programmer response

Check the entry point name is valid for the library module.

2183 (0887) (RC2183): MQRC_API_EXIT_LOAD_ERROR

Explanation

The API crossing exit module could not be linked to. If this message is returned when the API crossing exit is called *after* the process has been run, the process itself might have completed correctly.

Completion code

MQCC_FAILED

Programmer response

Ensure that the correct library concatenation has been specified, and that the API crossing exit module is executable and correctly named. Any uncommitted changes in a unit of work should be backed out. A unit of work that is coordinated by the queue manager is backed out automatically.

2184 (0888) (RC2184): MQRC_REMOTE_Q_NAME_ERROR

Explanation

On an MQOPEN or MQPUT1 call, one of the following occurred:

- A local definition of a remote queue (or an alias to one) was specified, but the **RemoteQName** attribute in the remote queue definition is entirely blank. Note that this error occurs even if the `XmitQName` in the definition is not blank.
- The `ObjectQMgrName` field in the object descriptor is not blank and not the name of the local queue manager, but the `ObjectName` field is blank.

Completion code

MQCC_FAILED

Programmer response

Alter the local definition of the remote queue and supply a valid remote queue name, or supply a nonblank `ObjectName` in the object descriptor, as appropriate.

This reason code is also used to identify the corresponding event message "Remote Queue Name Error" on page 228.

2185 (0889) (RC2185): MQRC_INCONSISTENT_PERSISTENCE

Explanation

An MQPUT call was issued to put a message in a group or a segment of a logical message, but the value specified or defaulted for the `Persistence` field in MQMD is not consistent with the current group and segment information retained by the queue manager for the queue handle. All messages in a group and all segments in a logical message must have the same value for persistence, that is, all must be persistent, or all must be nonpersistent.

If the current call specifies MQPMO_LOGICAL_ORDER, the call fails. If the current call does not specify MQPMO_LOGICAL_ORDER, but the previous MQPUT call for the queue handle did, the call succeeds with completion code MQCC_WARNING.

Completion code

MQCC_WARNING or MQCC_FAILED

Programmer response

Modify the application to ensure that the same value of persistence is used for all messages in the group, or all segments of the logical message.

2186 (088A) (RC2186): MQRC_GMO_ERROR

Explanation

On an MQGET call, the MQGMO structure is not valid, for one of the following reasons:

- The `StrucId` field is not MQGMO_STRUC_ID.
- The `Version` field specifies a value that is not valid or not supported.
- The parameter pointer is not valid. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)
- The queue manager cannot copy the changed structure to application storage, even though the call is successful. This can occur, for example, if the pointer points to read-only storage.

Completion code

MQCC_FAILED

Programmer response

Ensure that input fields in the MQGMO structure are set correctly.

z/OS 2187 (088B) (RC2187): MQRC_CICS_BRIDGE_RESTRICTION

Explanation

It is not permitted to issue MQI calls from user transactions that are run in an MQ/CICS bridge environment where the bridge exit also issues MQI calls. The MQI call fails. If it occurs in the bridge exit, it results in a transaction abend. If it occurs in the user transaction, it can result in a transaction abend.

This reason code occurs only on z/OS.

Completion code

MQCC_FAILED

Programmer response

The transaction cannot be run using the MQ/CICS bridge. Refer to the appropriate CICS manual for information about restrictions in the MQ/CICS bridge environment.

2188 (088C) (RC2188): MQRC_STOPPED_BY_CLUSTER_EXIT

Explanation

An MQOPEN, MQPUT, or MQPUT1 call was issued to open or put a message on a cluster queue, but the cluster workload exit rejected the call.

Completion code

MQCC_FAILED

Programmer response

Check the cluster workload exit to ensure that it has been written correctly. Determine why it rejected the call and correct the problem.

2189 (088D) (RC2189): MQRC_CLUSTER_RESOLUTION_ERROR

Explanation

An MQOPEN, MQPUT, or MQPUT1 call was issued to open or put a message on a cluster queue, but the queue definition could not be resolved correctly because a response was required from the repository manager but none was available.

This reason code can occur when running the REFRESH CLUSTER command. See Application issues seen when running REFRESH CLUSTER

Completion code

MQCC_FAILED

Programmer response

Check that the repository manager is operating and that the queue and channel definitions are correct.

Related reference

Return code= 2189 MQRC_CLUSTER_RESOLUTION_ERROR when trying to open a queue in the cluster

2190 (088E) (RC2190): MQRC_CONVERTED_STRING_TOO_BIG

Explanation

On an MQGET call with the MQGMO_CONVERT option included in the **GetMsgOpts** parameter, a string in a fixed-length field in the message expanded during data conversion and exceeded the size of the field. When this happens, the queue manager tries discarding trailing blank characters and characters following the first null character to make the string fit, but in this case there were insufficient characters that could be discarded.

This reason code can also occur for messages with a format name of MQFMT_IMS_VAR_STRING. When this happens, it indicates that the IMS variable string expanded such that its length exceeded the capacity of the 2 byte binary length field contained within the structure of the IMS variable string. (The queue manager never discards trailing blanks in an IMS variable string.)

The message is returned unconverted, with the **CompCode** parameter of the MQGET call set to MQCC_WARNING. If the message consists of several parts, each of which is described by its own character-set and encoding fields (for example, a message with format name MQFMT_DEAD_LETTER_HEADER), some parts might be converted and other parts not converted. However, the values returned in the various character-set and encoding fields always correctly describe the relevant message data.

This reason code does not occur if the string can be made to fit by discarding trailing blank characters.

Completion code

MQCC_WARNING

Programmer response

Check that the fields in the message contain the correct values, and that the character-set identifiers specified by the sender and receiver of the message are correct. If they are, the layout of the data in the message must be modified to increase the lengths of the field, or fields so that there is sufficient space to permit the string, or strings to expand when converted.

Multi 2191 (088F) (RC2191): MQRC_TMC_ERROR

Explanation

An MQPUT or MQPUT1 call was issued, but the message data contains an MQTMC2 structure that is not valid. Possible errors include the following:

- The `StrucId` field is not MQTMC_STRUC_ID.
- The `Version` field is not MQTMC_VERSION_2.
- The **BufferLength** parameter of the call has a value that is too small to accommodate the structure (the structure extends beyond the end of the message).

Completion code

MQCC_FAILED

Programmer response

Check that the fields in the structure are set correctly.

2192 (0890) (RC2192): MQRC_PAGESET_FULL

Explanation

Former name for MQRC_STORAGE_MEDIUM_FULL.

z/OS 2192 (0890) (RC2192): MQRC_STORAGE_MEDIUM_FULL

Explanation

An MQI call or command was issued to operate on an object, but the call failed because the external storage medium is full. One of the following applies:

- A page-set data set is full (nonshared queues only).
- A coupling-facility structure is full (shared queues only).

  A structure is full if either all ENTRYs or all ELEMENTs are in use.
- A coupling-facility is full. This situation can arise when the coupling facility structure is configured to use SCM storage (SCMMAXSIZE configured in CFRM policy) and messages are offloaded to SCM storage because the coupling facility structure has reached 90% threshold. Additional SCM use requires further augmented storage for the structure and there is insufficient storage in the coupling-facility to support this.
- The SMDS was full.

You can get this reason code when the page set or SMDS were expanding, but the space was not yet available. Check the messages in the job log to see the status of any expansion.

This reason code occurs only on z/OS.

Completion code

MQCC_FAILED

Programmer response

Check which queues contain messages and look for applications that might be filling the queues unintentionally. Be aware that the queue that has caused the page set or coupling-facility structure to become full is not necessarily the queue referenced by the MQI call that returned MQRC_STORAGE_MEDIUM_FULL.

Check that all of the usual server applications are operating correctly and processing the messages on the queues.

If the applications and servers are operating correctly, increase the number of server applications to cope with the message load, or request the system programmer to increase the size of the page-set data sets, coupling-facility structure, or SMDS.

For the situation of a full structure, issue the z/OS command /DISPLAY XCF,STRUCTURE,STRNAME=**structure-name**, to see information about INITSIZE, MAXSIZE, MINSIZE and ALLOWAUTOALT settings. The command also shows current space usage, so the number of elements and entries can be seen.

SupportPac MP16 contains information about these settings, and you can use the z/OSMF Sysplex Management service CFRM policy editor tool to help estimate the structure size needed. For more information, see CFRM Policy Editor and Sizer.

z/OS 2193 (0891) (RC2193): MQRC_PAGESET_ERROR

Explanation

An error was encountered with the page set while attempting to access it for a locally defined queue. This could be because the queue is on a page set that does not exist. A console message is issued that tells you the number of the page set in error. For example if the error occurred in the TEST job, and your user identifier is ABCDEFG, the message is:

```
CSQI041I CSQIALLC JOB TEST USER ABCDEFG HAD ERROR ACCESSING PAGE SET 27
```

If this reason code occurs while attempting to delete a dynamic queue with MQCLOSE, the dynamic queue has not been deleted.

This reason code occurs only on z/OS.

Completion code

MQCC_FAILED

Programmer response

Check that the storage class for the queue maps to a valid page set using the DISPLAY Q(xx) STGCLASS, DISPLAY STGCLASS(xx), and DISPLAY USAGE PSID commands. If you are unable to resolve the problem, notify the system programmer who should:

- Collect the following diagnostic information:
  - A description of the actions that led to the error
  - A listing of the application program being run at the time of the error
  - Details of the page sets defined for use by the queue manager
- Attempt to re-create the problem, and take a system dump immediately after the error occurs
- Contact IBM Support

2194 (0892) (RC2194): MQRC_NAME_NOT_VALID_FOR_TYPE

Explanation

An MQOPEN call was issued to open the queue manager definition, but the `ObjectName` field in the **`ObjDesc`** parameter is not blank.

Completion code

MQCC_FAILED

Programmer response

Ensure that the `ObjectName` field is set to blanks.

2195 (0893) (RC2195): MQRC_UNEXPECTED_ERROR

Explanation

The call was rejected because an unexpected error occurred.

Completion code

MQCC_FAILED

Programmer response

Check the application's parameter list to ensure, for example, that the correct number of parameters was passed, and that data pointers and storage keys are valid. If the problem cannot be resolved, contact your system programmer.

- z/OS On z/OS, check the joblog and logrec, and whether any information has been displayed on the console. If this error occurs on an MQCONN or MQCONNX call, check that the subsystem named is an active MQ subsystem. In particular, check that it is not a Db2 subsystem. If the problem cannot be resolved, rerun the application with a CSQSNAP DD card (if you have not already got a dump) and send the resulting dump to IBM.
- IBM i On IBM i, consult the FFST record to obtain more detail about the problem.
- Linux AIX On AIX and Linux, consult the FDC file to obtain more detail about the problem.

2196 (0894) (RC2196): MQRC_UNKNOWN_XMIT_Q

Explanation

On an MQOPEN or MQPUT1 call, a message is to be sent to a remote queue manager. The **ObjectName** or the **ObjectQMgrName** in the object descriptor specifies the name of a local definition of a remote queue (in the latter case queue manager aliasing is being used), but the **XmitQName** attribute of the definition is not blank and not the name of a locally-defined queue.

Completion code

MQCC_FAILED

Programmer response

Check the values specified for `ObjectName` and `ObjectQMgrName`. If these are correct, check the queue definitions.

This reason code is also used to identify the corresponding event message "Unknown Transmission Queue" on page 243.

2197 (0895) (RC2197): MQRC_UNKNOWN_DEF_XMIT_Q

Explanation

An MQOPEN or MQPUT1 call was issued specifying a remote queue as the destination. If a local definition of the remote queue was specified, or if a queue manager alias is being resolved, the **XmitQName** attribute in the local definition is blank.

Because there is no queue defined with the same name as the destination queue manager, the queue manager has attempted to use the default transmission queue. However, the name defined by the **DefXmitQName** queue manager attribute is not the name of a locally-defined queue.

Completion code

MQCC_FAILED

Programmer response

Correct the queue definitions, or the queue manager attribute.

This reason code is also used to identify the corresponding event message "Unknown Default Transmission Queue" on page 236.

2198 (0896) (RC2198): MQRC_DEF_XMIT_Q_TYPE_ERROR

Explanation

An MQOPEN or MQPUT1 call was issued specifying a remote queue as the destination. Either a local definition of the remote queue was specified, or a queue manager alias was being resolved, but in either case the **XmitQName** attribute in the local definition is blank.

Because there is no transmission queue defined with the same name as the destination queue manager, the local queue manager has attempted to use the default transmission queue. However, although there is a queue defined by the **DefXmitQName** queue manager attribute, it is not a local queue.

Completion code

MQCC_FAILED

Programmer response

Do one of the following:

- Specify a local transmission queue as the value of the **XmitQName** attribute in the local definition of the remote queue.
- Define a local transmission queue with a name that is the same as that of the remote queue manager.
- Specify a local transmission queue as the value of the **DefXmitQName** queue manager attribute.

See XmitQName for more information about transmission queue names.

This reason code is also used to identify the corresponding event message "Default Transmission Queue Type Error" on page 183.

2199 (0897) (RC2199): MQRC_DEF_XMIT_Q_USAGE_ERROR

Explanation

An MQOPEN or MQPUT1 call was issued specifying a remote queue as the destination. Either a local definition of the remote queue was specified, or a queue manager alias was being resolved, but in either case the **XmitQName** attribute in the local definition is blank.

Because there is no transmission queue defined with the same name as the destination queue manager, the local queue manager has attempted to use the default transmission queue. However, the queue defined by the **DefXmitQName** queue manager attribute does not have a **Usage** attribute of MQUS_TRANSMISSION.

This reason code is returned from MQOPEN or MQPUT1, if the queue manager's Default Transmission Queue is about to be used, but the name of this queue is SYSTEM.CLUSTER.TRANSMIT.QUEUE. This queue is reserved for clustering, so it is not valid to set the queue manager's Default Transmission Queue to this name.

Completion code

MQCC_FAILED

Programmer response

Do one of the following:

- Specify a local transmission queue as the value of the **XmitQName** attribute in the local definition of the remote queue.
- Define a local transmission queue with a name that is the same as that of the remote queue manager.

- Specify a different local transmission queue as the value of the **DefXmitQName** queue manager attribute.
- Change the **Usage** attribute of the **DefXmitQName** queue to MQUS_TRANSMISSION.

See XmitQName for more information about transmission queue names.

This reason code is also used to identify the corresponding event message "Default Transmission Queue Usage Error" on page 185.

z/OS 2201 (0899) (RC2201): MQRC_NAME_IN_USE

Explanation

An MQOPEN call was issued to create a dynamic queue, but a queue with the same name as the dynamic queue already exists. The existing queue is one that is logically deleted, but for which there are still one or more open handles. For more information, see Usage note 3 in MQCLOSE.

This reason code occurs only on z/OS.

Completion code

MQCC_FAILED

Programmer response

Either ensure that all handles for the previous dynamic queue are closed, or ensure that the name of the new queue is unique; see the description for reason code MQRC_OBJECT_ALREADY_EXISTS.

2202 (089A) (RC2202): MQRC_CONNECTION_QUIESCING

Explanation

This reason code is issued when the connection to the queue manager is in quiescing state, and an application issues one of the following calls:

- MQCONN or MQCONNX
- MQOPEN, with no connection established, or with MQOO_FAIL_IF_QUIESCING included in the **Options** parameter
- MQGET, with MQGMO_FAIL_IF_QUIESCING included in the `Options` field of the **GetMsgOpts** parameter
- MQPUT or MQPUT1, with MQPMO_FAIL_IF_QUIESCING included in the `Options` field of the **PutMsgOpts** parameter

MQRC_CONNECTION_QUIESCING is also issued by the message channel agent (MCA) when the queue manager is in quiescing state.

Completion code

MQCC_FAILED

Programmer response

The application should tidy up and terminate. Any uncommitted changes in a unit of work should be backed out.

2203 (089B) (RC2203): MQRC_CONNECTION_STOPPING

Explanation

This reason code is issued when the connection to the queue manager is shutting down, and the application issues an MQI call. No more message-queuing calls can be issued. For the MQGET call, if the MQGMO_WAIT option was specified, the wait is canceled.

Note that the MQRC_CONNECTION_BROKEN reason may be returned instead if, as a result of system scheduling factors, the queue manager shuts down before the call completes.

MQRC_CONNECTION_STOPPING is also issued by the message channel agent (MCA) when the queue manager is shutting down.

For MQ MQI client applications, it is possible that the call did complete successfully, even though this reason code is returned with a CompCode of MQCC_FAILED.

Completion code

MQCC_FAILED

Programmer response

The application should tidy up and terminate. Any uncommitted changes in a unit of work should be backed out. A unit of work that is coordinated by the queue manager is backed out automatically.

z/OS 2204 (089C) (RC2204): MQRC_ADAPTER_NOT_AVAILABLE

Explanation

This is issued only for CICS applications, if any call is issued and the CICS adapter (a Task Related User Exit) has been disabled, or has not been enabled.

This reason code occurs only on z/OS.

Completion code

MQCC_FAILED

Programmer response

The application should tidy up and terminate. Any uncommitted changes in a unit of work should be backed out. A unit of work that is coordinated by the queue manager is backed out automatically.

2206 (089E) (RC2206): MQRC_MSG_ID_ERROR

Explanation

An MQGET call was issued to retrieve a message using the message identifier as a selection criterion, but the call failed because selection by message identifier is not supported on this queue.

z/OS On z/OS, the queue is a shared queue, but the **IndexType** queue attribute does not have an appropriate value:

- If selection is by message identifier alone, **IndexType** must have the value MQIT_MSG_ID.
- If selection is by message identifier and correlation identifier combined, **IndexType** must have the value MQIT_MSG_ID or MQIT_CORREL_ID. However, the match-any values of MQCI_NONE and

MQMI_NONE respectively are exceptions to this rule, and result in the 2206 MQRC_MSG_ID_ERROR reason code.

Completion code

MQCC_FAILED

Programmer response

Do one of the following:

- Modify the application so that it does not use selection by message identifier: set the *MsgId* field to MQMI_NONE and do not specify MQMO_MATCH_MSG_ID in MQGMO.
- On z/OS, change the value of the **IndexType** queue attribute to MQIT_MSG_ID.

2207 (089F) (RC2207): MQRC_CORREL_ID_ERROR

Explanation

An MQGET call was issued to retrieve a message using the correlation identifier as a selection criterion, but the call failed because selection by correlation identifier is not supported on this queue.

z/OS On z/OS, the queue is a shared queue, but the **IndexType** queue attribute does not have an appropriate value:

- If selection is by correlation identifier alone, **IndexType** must have the value MQIT_CORREL_ID.
- If selection is by correlation identifier and message identifier combined, **IndexType** must have the value MQIT_CORREL_ID or MQIT_MSG_ID.

Completion code

MQCC_FAILED

Programmer response

Do one of the following:

- On z/OS, change the **IndexType** queue attribute to MQIT_CORREL_ID.
- Modify the application so that it does not use selection by correlation identifier: set the *CorrelId* field to MQCI_NONE and do not specify MQMO_MATCH_CORREL_ID in MQGMO.

2208 (08A0) (RC2208): MQRC_FILE_SYSTEM_ERROR

Explanation

An unexpected return code was received from the file system, in attempting to perform an operation on a queue.

Completion code

MQCC_FAILED

Programmer response

Check the file system definition for the queue that was being accessed. For a VSAM file, check that the control interval is large enough for the maximum message length allowed for the queue.

2209 (08A1) (RC2209): MQRC_NO_MSG_LOCKED

Explanation

An MQGET call was issued with the MQGMO_UNLOCK option, but no message was currently locked.

Completion code

MQCC_WARNING

Programmer response

Check that a message was locked by an earlier MQGET call with the MQGMO_LOCK option for the same handle, and that no intervening call has caused the message to become unlocked.

z/OS 2217 (08A9) (RC2217): MQRC_CONNECTION_NOT_AUTHORIZED

Explanation

This reason code occurs only on z/OS.

If the queue manager has been configured to use Advanced Message Security this reason code is returned if an error occurs in security processing.

This reason code might indicate a privacy security policy has been defined for the target queue that does not identify any recipients.

This reason code is also returned to CICS applications if the CICS subsystem is not authorized to connect to the queue manager.

Completion code

MQCC_FAILED

Programmer response

Ensure that the subsystem is authorized to connect to the queue manager.

If you are using Advanced Message Security, check the queue manager and AMS task error logs. You should also check the job log of the connecting task for error messages.

2218 (08AA) (RC2218): MQRC_MSG_TOO_BIG_FOR_CHANNEL

Explanation

A message was put to a remote queue, but the message is larger than the maximum message length allowed by the channel. This reason code is returned in the Feedback field in the message descriptor of a report message.

Completion code

MQCC_FAILED

Programmer response

Check the channel definitions. Increase the maximum message length that the channel can accept, or break the message into several smaller messages.

2219 (08AB) (RC2219): MQRC_CALL_IN_PROGRESS

Explanation

The application issued an MQI call whilst another MQI call was already being processed for that connection. Only one call per application connection can be processed at a time.

Concurrent calls can arise when an application uses multiple threads, or when an exit is invoked as part of the processing of an MQI call. For example, a data-conversion exit invoked as part of the processing of the MQGET call may try to issue an MQI call.

- z/OS On z/OS, concurrent calls can arise only with batch or IMS applications; an example is when a subtask ends while an MQI call is in progress (for example, an MQGET that is waiting), and there is an end-of-task exit routine that issues another MQI call.
- Windows On Windows, concurrent calls can also arise if an MQI call is issued in response to a user message while another MQI call is in progress.
- If the application is using multiple threads with shared handles, MQRC_CALL_IN_PROGRESS occurs when the handle specified on the call is already in use by another thread and MQCNO_HANDLE_SHARE_NO_BLOCK was specified on the MQCONNX call.

Completion code

MQCC_FAILED

Programmer response

Ensure that an MQI call cannot be issued while another one is active. Do not issue MQI calls from within a data-conversion exit.

z/OS On z/OS, if you want to provide a subtask to allow an application that is waiting for a message to arrive to be canceled, wait for the message by using MQGET with MQGMO_SET_SIGNAL, rather than MQGMO_WAIT.

2220 (08AC) (RC2220): MQRC_RMH_ERROR

Explanation

An MQPUT or MQPUT1 call was issued, but the message data contains an MQRMH structure that is not valid. Possible errors include the following:

- The *StrucId* field is not MQRMH_STRUC_ID.
- The *Version* field is not MQRMH_VERSION_1.
- The *StrucLength* field specifies a value that is too small to include the structure plus the variable-length data at the end of the structure.
- The *CodedCharSetId* field is zero, or a negative value that is not valid.
- The **BufferLength** parameter of the call has a value that is too small to accommodate the structure (the structure extends beyond the end of the message).

Completion code

MQCC_FAILED

Programmer response

Check that the fields in the structure are set correctly. Ensure that the application sets the `CodedCharSetId` field to a valid value (note: MQCCSI_DEFAULT, MQCCSI_EMBEDDED, MQCCSI_Q_MGR, and MQCCSI_UNDEFINED are not valid in this field).

2222 (08AE) (RC2222): MQRC_Q_MGR_ACTIVE

Explanation

This condition is detected when a queue manager becomes active.

z/OS On z/OS, this event is not generated for the first start of a queue manager, only on subsequent restarts.

Completion code

MQCC_WARNING

Programmer response

None. This reason code is only used to identify the corresponding event message "Queue Manager Active" on page 216.

2223 (08AF) (RC2223): MQRC_Q_MGR_NOT_ACTIVE

Explanation

This condition is detected when a queue manager is requested to stop or quiesce.

Completion code

MQCC_WARNING

Programmer response

None. This reason code is only used to identify the corresponding event message "Queue Manager Not Active" on page 217.

2224 (08B0) (RC2224): MQRC_Q_DEPTH_HIGH

Explanation

An MQPUT or MQPUT1 call has caused the queue depth to be incremented to, or greater than, the limit specified in the **QDepthHighLimit** attribute.

Completion code

MQCC_WARNING

Programmer response

None. This reason code is only used to identify the corresponding event message "Queue Depth High" on page 212.

2225 (08B1) (RC2225): MQRC_Q_DEPTH_LOW

Explanation

An MQGET call has caused the queue depth to be decremented to, or less than, the limit specified in the **QDepthLowLimit** attribute.

Completion code

MQCC_WARNING

Programmer response

None. This reason code is only used to identify the corresponding event message "Queue Depth Low" on page 213.

2226 (08B2) (RC2226): MQRC_Q_SERVICE_INTERVAL_HIGH

Explanation

No successful gets or puts have been detected within an interval that is greater than the limit specified in the **QServiceInterval** attribute.

Completion code

MQCC_WARNING

Programmer response

None. This reason code is only used to identify the corresponding event message "Queue Service Interval High" on page 218.

2227 (08B3) (RC2227): MQRC_Q_SERVICE_INTERVAL_OK

Explanation

A successful get has been detected within an interval that is less than or equal to the limit specified in the **QServiceInterval** attribute.

Completion code

MQCC_WARNING

Programmer response

None. This reason code is only used to identify the corresponding event message "Queue Service Interval OK" on page 219.

2228 (08B4) (RC2228): MQRC_RFH_HEADER_FIELD_ERROR

Explanation

An expected RFH header field was not found or had an invalid value.

Completion code

MQCC_FAILED

Programmer response

Check the associated error message, and ensure that the RFH2 section of the IBM MQ request message contains all the mandatory fields, and that these fields have valid values.

2229 (08B5) (RC2229): MQRC_RAS_PROPERTY_ERROR

Explanation

There is an error related to the RAS property file. The file might be missing, it might be not accessible, or the commands in the file might be incorrect.

Completion code

MQCC_FAILED

Programmer response

Look at the associated error message, which explains the error in detail. Correct the error and try again.

2232 (08B8) (RC2232): MQRC_UNIT_OF_WORK_NOT_STARTED

Explanation

An MQGET, MQPUT or MQPUT1 call was issued to get or put a message within a unit of work, but no TM/MP transaction had been started. If MQGMO_NO_SYNCPOINT is not specified on MQGET, or MQPMO_NO_SYNCPOINT is not specified on MQPUT or MQPUT1 (the default), the call requires a unit of work.

Completion code

MQCC_FAILED

Programmer response

Ensure a TM/MP transaction is available, or issue the MQGET call with the MQGMO_NO_SYNCPOINT option, or the MQPUT or MQPUT1 call with the MQPMO_NO_SYNCPOINT option, which will cause a transaction to be started automatically.

Multi 2233 (08B9) (RC2233): MQRC_CHANNEL_AUTO_DEF_OK

Explanation

This condition is detected when the automatic definition of a channel is successful. The channel is defined by the MCA.

Completion code

MQCC_WARNING

Programmer response

None. This reason code is only used to identify the corresponding event message "Channel Auto-definition OK" on page 153.

Multi 2234 (08BA) (RC2234): MQRC_CHANNEL_AUTO_DEF_ERROR

Explanation

This condition is detected when the automatic definition of a channel fails; this might be because an error occurred during the definition process, or because the channel automatic-definition exit inhibited the definition. Additional information is returned in the event message indicating the reason for the failure.

Completion code

MQCC_WARNING

Programmer response

This reason code is only used to identify the corresponding event message "Channel Auto-definition Error" on page 151.

Examine the additional information returned in the event message to determine the reason for the failure.

Multi 2235 (08BB) (RC2235): MQRC_CFH_ERROR

Explanation

An MQPUT or MQPUT1 call was issued, but the message data contains an MQCFH structure that is not valid.

Completion code

MQCC_FAILED

Programmer response

Check that the fields in the structure are set correctly.

Multi 2236 (08BC) (RC2236): MQRC_CFIL_ERROR

Explanation

An MQPUT or MQPUT1 call was issued, but the message data contains an MQCFIL or MQRCFIL64 structure that is not valid.

Completion code

MQCC_FAILED

Programmer response

Check that the fields in the structure are set correctly.

Multi 2237 (08BD) (RC2237): MQRC_CFIN_ERROR

Explanation

An MQPUT or MQPUT1 call was issued, but the message data contains an MQCFIN or MQCFIN64 structure that is not valid.

Completion code

MQCC_FAILED

Programmer response

Check that the fields in the structure are set correctly.

Multi 2238 (08BE) (RC2238): MQRC_CFSL_ERROR

Explanation

An MQPUT or MQPUT1 call was issued, but the message data contains an MQCFSL structure that is not valid.

Completion code

MQCC_FAILED

Programmer response

Check that the fields in the structure are set correctly.

Multi 2239 (08BF) (RC2239): MQRC_CFST_ERROR

Explanation

An MQPUT or MQPUT1 call was issued, but the message data contains an MQCFST structure that is not valid.

Completion code

MQCC_FAILED

Programmer response

Check that the fields in the structure are set correctly.

2241 (08C1) (RC2241): MQRC_INCOMPLETE_GROUP

Explanation

An operation was attempted on a queue using a queue handle that had an incomplete message group. This reason code can arise in the following situations:

- On the MQPUT call, when the application specifies MQPMO_LOGICAL_ORDER and attempts to put a message that is not in a group. The completion code is MQCC_FAILED in this case.

- On the MQPUT call, when the application does not specify MQPMO_LOGICAL_ORDER, but the previous MQPUT call for the queue handle did specify MQPMO_LOGICAL_ORDER. The completion code is MQCC_WARNING in this case.
- On the MQGET call, when the application does not specify MQGMO_LOGICAL_ORDER, but the previous MQGET call for the queue handle did specify MQGMO_LOGICAL_ORDER. The completion code is MQCC_WARNING in this case.
- On the MQCLOSE call, when the application attempts to close the queue that has the incomplete message group. The completion code is MQCC_WARNING in this case.

If there is an incomplete logical message as well as an incomplete message group, reason code MQRC_INCOMPLETE_MSG is returned in preference to MQRC_INCOMPLETE_GROUP.

Completion code

MQCC_WARNING or MQCC_FAILED

Programmer response

If this reason code is expected, no corrective action is required. Otherwise, ensure that the MQPUT call for the last message in the group specifies MQMF_LAST_MSG_IN_GROUP.

2242 (08C2) (RC2242): MQRC_INCOMPLETE_MSG

Explanation

An operation was attempted on a queue using a queue handle that had an incomplete logical message. This reason code can arise in the following situations:

- On the MQPUT call, when the application specifies MQPMO_LOGICAL_ORDER and attempts to put a message that is not a segment, or that has a setting for the MQMF_LAST_MSG_IN_GROUP flag that is different from the previous message. The completion code is MQCC_FAILED in this case.
- On the MQPUT call, when the application does not specify MQPMO_LOGICAL_ORDER, but the previous MQPUT call for the queue handle did specify MQPMO_LOGICAL_ORDER. The completion code is MQCC_WARNING in this case.
- On the MQGET call, when the application does not specify MQGMO_LOGICAL_ORDER, but the previous MQGET call for the queue handle did specify MQGMO_LOGICAL_ORDER. The completion code is MQCC_WARNING in this case.
- On the MQCLOSE call, when the application attempts to close the queue that has the incomplete logical message. The completion code is MQCC_WARNING in this case.

Completion code

MQCC_WARNING or MQCC_FAILED

Programmer response

If this reason code is expected, no corrective action is required. Otherwise, ensure that the MQPUT call for the last segment specifies MQMF_LAST_SEGMENT.

Multi 2243 (08C3) (RC2243): MQRC_INCONSISTENT_CCSIDS

Explanation

An MQGET call was issued specifying the MQGMO_COMPLETE_MSG option, but the message to be retrieved consists of two or more segments that have differing values for the *CodedCharSetId* field

in MQMD. This can arise when the segments take different paths through the network, and some of those paths have MCA sender conversion enabled. The call succeeds with a completion code of MQCC_WARNING, but only the first few segments that have identical character-set identifiers are returned.

Completion code

MQCC_WARNING

Programmer response

Remove the MQGMO_COMPLETE_MSG option from the MQGET call and retrieve the remaining message segments one by one.

Multi 2244 (08C4) (RC2244): MQRC_INCONSISTENT_ENCODINGS

Explanation

An MQGET call was issued specifying the MQGMO_COMPLETE_MSG option, but the message to be retrieved consists of two or more segments that have differing values for the *Encoding* field in MQMD. This can arise when the segments take different paths through the network, and some of those paths have MCA sender conversion enabled. The call succeeds with a completion code of MQCC_WARNING, but only the first few segments that have identical encodings are returned.

Completion code

MQCC_WARNING

Programmer response

Remove the MQGMO_COMPLETE_MSG option from the MQGET call and retrieve the remaining message segments one by one.

2245 (08C5) (RC2245): MQRC_INCONSISTENT_UOW

Explanation

One of the following applies:

- An MQPUT call was issued to put a message in a group or a segment of a logical message, but the value specified or defaulted for the MQPMO_SYNCPOINT option is not consistent with the current group and segment information retained by the queue manager for the queue handle.

  If the current call specifies MQPMO_LOGICAL_ORDER, the call fails. If the current call does not specify MQPMO_LOGICAL_ORDER, but the previous MQPUT call for the queue handle did, the call succeeds with completion code MQCC_WARNING.
- An MQGET call was issued to remove from the queue a message in a group or a segment of a logical message, but the value specified or defaulted for the MQGMO_SYNCPOINT option is not consistent with the current group and segment information retained by the queue manager for the queue handle.

  If the current call specifies MQGMO_LOGICAL_ORDER, the call fails. If the current call does not specify MQGMO_LOGICAL_ORDER, but the previous MQGET call for the queue handle did, the call succeeds with completion code MQCC_WARNING.

Completion code

MQCC_WARNING or MQCC_FAILED

Programmer response

Modify the application to ensure that the same unit-of-work specification is used for all messages in the group, or all segments of the logical message.

Multi 2246 (08C6) (RC2246): MQRC_INVALID_MSG_UNDER_CURSOR

Explanation

An MQGET call was issued specifying the MQGMO_COMPLETE_MSG option with either MQGMO_MSG_UNDER_CURSOR or MQGMO_BROWSE_MSG_UNDER_CURSOR, but the message that is under the cursor has an MQMD with an *Offset* field that is greater than zero. Because MQGMO_COMPLETE_MSG was specified, the message is not valid for retrieval.

Completion code

MQCC_FAILED

Programmer response

Reposition the browse cursor so that it is located on a message with an *Offset* field in MQMD that is zero. Alternatively, remove the MQGMO_COMPLETE_MSG option.

2247 (08C7) (RC2247): MQRC_MATCH_OPTIONS_ERROR

Explanation

An MQGET call was issued, but the value of the `MatchOptions` field in the **GetMsgOpts** parameter is not valid, for one of the following reasons:

- An undefined option is specified.
- All of the following statements are true:
  - MQGMO_LOGICAL_ORDER is specified.
  - There is a current message group or logical message for the queue handle.
  - Neither MQGMO_BROWSE_MSG_UNDER_CURSOR nor MQGMO_MSG_UNDER_CURSOR is specified.
  - One or more of the MQMO_* options is specified.
  - The values of the fields in the **MsgDesc** parameter corresponding to the MQMO_* options specified, differ from the values of those fields in the MQMD for the message to be returned next.
- z/OS On z/OS, one or more of the options specified is not valid for the index type of the queue.

Completion code

MQCC_FAILED

Programmer response

Ensure that only valid options are specified for the field.

2248 (08C8) (RC2248): MQRC_MDE_ERROR

Explanation

An MQPUT or MQPUT1 call was issued, but the message data contains an MQMDE structure that is not valid. Possible errors include the following:

- The `StrucId` field is not MQMDE_STRUC_ID.
- The `Version` field is not MQMDE_VERSION_2.
- The `StrucLength` field is not MQMDE_LENGTH_2.
- The `CodedCharSetId` field is zero, or a negative value that is not valid.
- The **`BufferLength`** parameter of the call has a value that is too small to accommodate the structure (the structure extends beyond the end of the message).

Completion code

MQCC_FAILED

Programmer response

Check that the fields in the structure are set correctly. Ensure that the application sets the `CodedCharSetId` field to a valid value (note: MQCCSI_DEFAULT, MQCCSI_EMBEDDED, MQCCSI_Q_MGR, and MQCCSI_UNDEFINED are not valid in this field).

2249 (08C9) (RC2249): MQRC_MSG_FLAGS_ERROR

Explanation

An MQPUT or MQPUT1 call was issued, but the `MsgFlags` field in the message descriptor MQMD contains one or more message flags that are not recognized by the local queue manager. The message flags that cause this reason code to be returned depend on the destination of the message; see the description of REPORT in Report options and message flags for more information.

This reason code can also occur in the `Feedback` field in the MQMD of a report message, or in the `Reason` field in the MQDLH structure of a message on the dead-letter queue; in both cases it indicates that the destination queue manager does not support one or more of the message flags specified by the sender of the message.

Completion code

MQCC_FAILED

Programmer response

Do the following:

- Ensure that the `MsgFlags` field in the message descriptor is initialized with a value when the message descriptor is declared, or is assigned a value prior to the MQPUT or MQPUT1 call. Specify MQMF_NONE if no message flags are needed.
- Ensure that the message flags specified are valid; see the `MsgFlags` field described in the description of MQMD in MsgFlags (MQLONG) for valid message flags.
- If multiple message flags are being set by adding the individual message flags together, ensure that the same message flag is not added twice.
- z/OS On z/OS, ensure that the message flags specified are valid for the index type of the queue; see the description of the `MsgFlags` field in MQMD for further details.

2250 (08CA) (RC2250): MQRC_MSG_SEQ_NUMBER_ERROR

Explanation

An MQGET, MQPUT, or MQPUT1 call was issued, but the value of the `MsgSeqNumber` field in the MQMD or MQMDE structure is less than one or greater than 999 999 999.

This error can also occur on the MQPUT call if the `MsgSeqNumber` field would have become greater than 999 999 999 as a result of the call.

Completion code

MQCC_FAILED

Programmer response

Specify a value in the range 1 through 999 999 999. Do not attempt to create a message group containing more than 999 999 999 messages.

2251 (08CB) (RC2251): MQRC_OFFSET_ERROR

Explanation

An MQPUT or MQPUT1 call was issued, but the value of the `Offset` field in the MQMD or MQMDE structure is less than zero or greater than 999 999 999.

This error can also occur on the MQPUT call if the `Offset` field would have become greater than 999 999 999 as a result of the call.

Completion code

MQCC_FAILED

Programmer response

Specify a value in the range 0 through 999 999 999. Do not attempt to create a message segment that would extend beyond an offset of 999 999 999.

2252 (08CC) (RC2252): MQRC_ORIGINAL_LENGTH_ERROR

Explanation

An MQPUT or MQPUT1 call was issued to put a report message that is a segment, but the `OriginalLength` field in the MQMD or MQMDE structure is either:

- Less than the length of data in the message, or
- Less than one (for a segment that is not the last segment), or
- Less than zero (for a segment that is the last segment)

Completion code

MQCC_FAILED

Programmer response

Specify a value that is greater than zero. Zero is valid only for the last segment.

Multi 2253 (08CD) (RC2253): MQRC_SEGMENT_LENGTH_ZERO

Explanation

An MQPUT or MQPUT1 call was issued to put the first or an intermediate segment of a logical message, but the length of the application message data in the segment (excluding any MQ headers that may be present) is zero. The length must be at least one for the first or intermediate segment.

Completion code

MQCC_FAILED

Programmer response

Check the application logic to ensure that segments are put with a length of one or greater. Only the last segment of a logical message is permitted to have a length of zero.

Multi 2255 (08CF) (RC2255): MQRC_UOW_NOT_AVAILABLE

Explanation

An MQGET, MQPUT, or MQPUT1 call was issued to get or put a message outside a unit of work, but the options specified on the call required the queue manager to process the call within a unit of work. Because there is already a user-defined unit of work in existence, the queue manager was unable to create a temporary unit of work for the duration of the call.

This reason occurs in the following circumstances:

- On an MQGET call, when the MQGMO_COMPLETE_MSG option is specified in MQGMO and the logical message to be retrieved is persistent and consists of two or more segments.
- On an MQPUT or MQPUT1 call, when the MQMF_SEGMENTATION_ALLOWED flag is specified in MQMD and the message requires segmentation.

Completion code

MQCC_FAILED

Programmer response

Issue the MQGET, MQPUT, or MQPUT1 call inside the user-defined unit of work. Alternatively, for the MQPUT or MQPUT1 call, reduce the size of the message so that it does not require segmentation by the queue manager.

2256 (08D0) (RC2256): MQRC_WRONG_GMO_VERSION

Explanation

An MQGET call was issued specifying options that required an MQGMO with a version number not less than MQGMO_VERSION_2, but the MQGMO supplied did not satisfy this condition.

Completion code

MQCC_FAILED

Programmer response

Modify the application to pass a version-2 MQGMO. Check the application logic to ensure that the `Version` field in MQGMO has been set to MQGMO_VERSION_2. Alternatively, remove the option that requires the version-2 MQGMO.

2257 (08D1) (RC2257): MQRC_WRONG_MD_VERSION

Explanation

An MQGET, MQPUT, or MQPUT1 call was issued specifying options that required an MQMD with a version number not less than MQMD_VERSION_2, but the MQMD supplied did not satisfy this condition.

Completion code

MQCC_FAILED

Programmer response

Modify the application to pass a version-2 MQMD. Check the application logic to ensure that the `Version` field in MQMD has been set to MQMD_VERSION_2. Alternatively, remove the option that requires the version-2 MQMD.

Multi 2258 (08D2) (RC2258): MQRC_GROUP_ID_ERROR

Explanation

An MQPUT or MQPUT1 call was issued to put a distribution-list message that is also a message in a group, a message segment, or has segmentation allowed, but an invalid combination of options and values was specified. All of the following statements are true:

- MQPMO_LOGICAL_ORDER is not specified in the `Options` field in MQPMO.
- Either there are too few MQPMR records provided by MQPMO, or the `GroupId` field is not present in the MQPMR records.
- One or more of the following flags is specified in the `MsgFlags` field in MQMD or MQMDE:
  - MQMF_SEGMENTATION_ALLOWED
  - MQMF_*_MSG_IN_GROUP
  - MQMF_*_SEGMENT
- The *GroupId* field in MQMD or MQMDE is not MQGI_NONE.

This combination of options and values would result in the same group identifier being used for all of the destinations in the distribution list; this is not permitted by the queue manager.

Completion code

MQCC_FAILED

Programmer response

Specify MQGI_NONE for the `GroupId` field in MQMD or MQMDE. Alternatively, if the call is MQPUT specify MQPMO_LOGICAL_ORDER in the `Options` field in MQPMO.

Multi 2259 (08D3) (RC2259): MQRC_INCONSISTENT_BROWSE

Explanation

An MQGET call was issued with the MQGMO_BROWSE_NEXT option specified, but the specification of the MQGMO_LOGICAL_ORDER option for the call is different from the specification of that option for the previous call for the queue handle. Either both calls must specify MQGMO_LOGICAL_ORDER, or neither call must specify MQGMO_LOGICAL_ORDER.

Completion code

MQCC_FAILED

Programmer response

Add or remove the MQGMO_LOGICAL_ORDER option as appropriate. Alternatively, to switch between logical order and physical order, specify the MQGMO_BROWSE_FIRST option to restart the scan from the beginning of the queue, omitting or specifying MQGMO_LOGICAL_ORDER as required.

2260 (08D4) (RC2260): MQRC_XQH_ERROR

Explanation

An MQPUT or MQPUT1 call was issued, but the message data contains an MQXQH structure that is not valid. Possible errors include the following:

- The `StrucId` field is not MQXQH_STRUC_ID.
- The `Version` field is not MQXQH_VERSION_1.
- The **`BufferLength`** parameter of the call has a value that is too small to accommodate the structure (the structure extends beyond the end of the message).

Completion code

MQCC_FAILED

Programmer response

Check that the fields in the structure are set correctly.

Multi 2261 (08D5) (RC2261): MQRC_SRC_ENV_ERROR

Explanation

This reason occurs when a channel exit that processes reference messages detects an error in the source environment data of a reference message header (MQRMH). One of the following is true:

- `SrcEnvLength` is less than zero.
- `SrcEnvLength` is greater than zero, but there is no source environment data.
- `SrcEnvLength` is greater than zero, but `SrcEnvOffset` is negative, zero, or less than the length of the fixed part of MQRMH.
- `SrcEnvLength` is greater than zero, but `SrcEnvOffset` plus `SrcEnvLength` is greater than `StrucLength`.

The exit returns this reason in the `Feedback` field of the MQCXP structure. If an exception report is requested, it is copied to the `Feedback` field of the MQMD associated with the report.

Completion code

MQCC_FAILED

Programmer response

Specify the source environment data correctly.

Multi 2262 (08D6) (RC2262): MQRC_SRC_NAME_ERROR

Explanation

This reason occurs when a channel exit that processes reference messages detects an error in the source name data of a reference message header (MQRMH). One of the following is true:

- `SrcNameLength` is less than zero.
- `SrcNameLength` is greater than zero, but there is no source name data.
- `SrcNameLength` is greater than zero, but `SrcNameOffset` is negative, zero, or less than the length of the fixed part of MQRMH.
- `SrcNameLength` is greater than zero, but `SrcNameOffset` plus `SrcNameLength` is greater than `StrucLength`.

The exit returns this reason in the `Feedback` field of the MQCXP structure. If an exception report is requested, it is copied to the `Feedback` field of the MQMD associated with the report.

Completion code

MQCC_FAILED

Programmer response

Specify the source name data correctly.

Multi 2263 (08D7) (RC2263): MQRC_DEST_ENV_ERROR

Explanation

This reason occurs when a channel exit that processes reference messages detects an error in the destination environment data of a reference message header (MQRMH). One of the following is true:

- `DestEnvLength` is less than zero.
- `DestEnvLength` is greater than zero, but there is no destination environment data.
- `DestEnvLength` is greater than zero, but `DestEnvOffset` is negative, zero, or less than the length of the fixed part of MQRMH.
- `DestEnvLength` is greater than zero, but `DestEnvOffset` plus `DestEnvLength` is greater than `StrucLength`.

The exit returns this reason in the `Feedback` field of the MQCXP structure. If an exception report is requested, it is copied to the `Feedback` field of the MQMD associated with the report.

Completion code

MQCC_FAILED

Programmer response

Specify the destination environment data correctly.

Multi 2264 (08D8) (RC2264): MQRC_DEST_NAME_ERROR

Explanation

This reason occurs when a channel exit that processes reference messages detects an error in the destination name data of a reference message header (MQRMH). One of the following is true:

- `DestNameLength` is less than zero.
- `DestNameLength` is greater than zero, but there is no destination name data.
- `DestNameLength` is greater than zero, but `DestNameOffset` is negative, zero, or less than the length of the fixed part of MQRMH.
- `DestNameLength` is greater than zero, but `DestNameOffset` plus `DestNameLength` is greater than `StrucLength`.

The exit returns this reason in the `Feedback` field of the MQCXP structure. If an exception report is requested, it is copied to the `Feedback` field of the MQMD associated with the report.

Completion code

MQCC_FAILED

Programmer response

Specify the destination name data correctly.

2265 (08D9) (RC2265): MQRC_TM_ERROR

Explanation

An MQPUT or MQPUT1 call was issued, but the message data contains an MQTM structure that is not valid. Possible errors include the following:

- The `StrucId` field is not MQTM_STRUC_ID.
- The `Version` field is not MQTM_VERSION_1.
- The **`BufferLength`** parameter of the call has a value that is too small to accommodate the structure (the structure extends beyond the end of the message).

Completion code

MQCC_FAILED

Programmer response

Check that the fields in the structure are set correctly.

2266 (08DA) (RC2266): MQRC_CLUSTER_EXIT_ERROR

Explanation

An MQOPEN, MQPUT, or MQPUT1 call was issued to open or put a message on a cluster queue, but the cluster workload exit defined by the queue manager's **`ClusterWorkloadExit`** attribute failed

unexpectedly or did not respond in time. Subsequent MQOPEN, MQPUT, and MQPUT1 calls for this queue handle are processed as though the **ClusterWorkloadExit** attribute were blank.

z/OS On z/OS, a message giving more information about the error is written to the system log, for example message CSQV455E or CSQV456E.

Completion code

MQCC_FAILED

Programmer response

Check the cluster workload exit to ensure that it has been written correctly.

2267 (08DB) (RC2267): MQRC_CLUSTER_EXIT_LOAD_ERROR

Explanation

An MQCONN or MQCONNX call was issued to connect to a queue manager, but the queue manager was unable to load the cluster workload exit. Execution continues without the cluster workload exit.

z/OS On z/OS, if the cluster workload exit cannot be loaded, a message is written to the system log, for example message CSQV453I. Processing continues as though the **ClusterWorkloadExit** attribute had been blank.

Completion code

MQCC_WARNING

Programmer response

Ensure that the queue manager's **ClusterWorkloadExit** attribute has the correct value, and that the exit has been installed into the correct location.

2268 (08DC) (RC2268): MQRC_CLUSTER_PUT_INHIBITED

Explanation

An MQOPEN call with the MQOO_OUTPUT and MQOO_BIND_ON_OPEN options in effect was issued for a cluster queue, but the call failed because all of the following statements are true:

- All instances of the cluster queue are currently put-inhibited, that is, all of the queue instances have the **InhibitPut** attribute set to MQQA_PUT_INHIBITED.
- There is no local instance of the queue. (If there is a local instance, the MQOPEN call succeeds, even if the local instance is put-inhibited.)
- There is no cluster workload exit for the queue, or there is a cluster workload exit but it did not choose a queue instance. (If the cluster workload exit does choose a queue instance, the MQOPEN call succeeds, even if that instance is put-inhibited.)

If the MQOO_BIND_NOT_FIXED option is specified on the MQOPEN call, the call can succeed even if all of the queues in the cluster are put-inhibited. However, a subsequent MQPUT call might fail if all of the queues are still put-inhibited at the time of the MQPUT call.

Completion code

MQCC_FAILED

Programmer response

If the system design allows put requests to be inhibited for short periods, retry the operation later. If the problem persists, determine why all of the queues in the cluster are put-inhibited.

2269 (08DD) (RC2269): MQRC_CLUSTER_RESOURCE_ERROR

Explanation

An MQOPEN, MQPUT, or MQPUT1 call was issued for a cluster queue, but an error occurred whilst trying to use a resource required for clustering.

Completion code

MQCC_FAILED

Programmer response

Do the following:

- Check that the SYSTEM.CLUSTER.* queues are not put inhibited or full.
- Check the event queues for any events relating to the SYSTEM.CLUSTER.* queues, as these may give guidance as to the nature of the failure.
- Check that the repository queue manager is available.
- z/OS On z/OS, check the console for signs of the failure, such as full page sets.

2270 (08DE) (RC2270): MQRC_NO_DESTINATIONS_AVAILABLE

Explanation

An MQPUT or MQPUT1 call was issued to put a message on a cluster queue, but at the time of the call there were no longer any instances of the queue in the cluster. The message therefore could not be sent.

This situation can occur when MQOO_BIND_NOT_FIXED is specified on the MQOPEN call that opens the queue, or MQPUT1 is used to put the message.

This reason code can also occur when running the REFRESH CLUSTER command. See Application issues seen when running REFRESH CLUSTER

Completion code

MQCC_FAILED

Programmer response

Check the queue definition and queue status to determine why all instances of the queue were removed from the cluster. Correct the problem and rerun the application.

z/OS 2271 (08DF) (RC2271): MQRC_CONN_TAG_IN_USE

Explanation

An MQCONNX call was issued specifying one of the MQCNO_*_CONN_TAG_* options, but the call failed because the connection tag specified by ConnTag in MQCNO is in use by an active process or thread, or there is an unresolved unit of work that references this connection tag.

This reason code occurs only on z/OS.

Completion code

MQCC_FAILED

Programmer response

The problem is likely to be transitory. The application should wait a short while and then retry the operation.

2272 (08E0) (RC2272): MQRC_PARTIALLY_CONVERTED

Explanation

On an MQGET call with the MQGMO_CONVERT option included in the **GetMsgOpts** parameter, one or more MQ header structures in the message data could not be converted to the specified target character set or encoding. In this situation, the MQ header structures are converted to the queue manager's character set and encoding, and the application data in the message is converted to the target character set and encoding. On return from the call, the values returned in the various `CodedCharSetId` and `Encoding` fields in the **MsgDesc** parameter and MQ header structures indicate the character set and encoding that apply to each part of the message. The call completes with MQCC_WARNING.

This reason code usually occurs when the specified target character set is one that causes the character strings in the MQ header structures to expand beyond the lengths of their fields. Unicode character set UTF-16 is an example of a character set that causes this to happen.

Completion code

MQCC_FAILED

Programmer response

If this is an expected situation, no corrective action is required.

If this is an unexpected situation, check that the MQ header structures contain valid data. If they do, specify as the target character set a character set that does not cause the strings to expand.

2273 (08E1) (RC2273): MQRC_CONNECTION_ERROR

Explanation

An MQCONN or MQCONNX call failed for one of the following reasons:

- The installation and customization options chosen for IBM MQ do not allow connection by the type of application being used.
- The system parameter module is not at the same release level as the queue manager.
- The channel initiator is not at the same release level as the queue manager.
- An internal error was detected by the queue manager.

Completion code

MQCC_FAILED

Programmer response

None, if the installation and customization options chosen for IBM MQ do not allow all functions to be used.

Otherwise, if this occurs while starting the channel initiator, ensure that the queue manager and the channel initiator are both at the same release level and that their started task JCL procedures both specify the same level of IBM MQ program libraries; if this occurs while starting the queue manager, relinkedit the system parameter module (CSQZPARM) to ensure that it is at the correct level. If the problem persists, contact IBM Support.

z/OS 2274 (08E2) (RC2274): MQRC_OPTION_ENVIRONMENT_ERROR

Explanation

An MQGET call with the MQGMO_MARK_SKIP_BACKOUT option specified was issued from a Db2 Stored Procedure. The call failed because the MQGMO_MARK_SKIP_BACKOUT option cannot be used from a Db2 Stored Procedure.

This reason code occurs only on z/OS.

Completion code

MQCC_FAILED

Programmer response

Remove the MQGMO_MARK_SKIP_BACKOUT option from the MQGET call.

Multi 2277 (08E5) (RC2277): MQRC_CD_ERROR

Explanation

An MQCONNX call was issued to connect to a queue manager, but the MQCD channel definition structure addressed by the *ClientConnOffset* or *ClientConnPtr* field in MQCNO contains data that is not valid. Consult the error log for more information about the nature of the error.

Completion code

MQCC_FAILED

Programmer response

Ensure that input fields in the MQCD structure are set correctly.

Multi 2278 (08E6) (RC2278): MQRC_CLIENT_CONN_ERROR

Explanation

An MQCONNX call was issued to connect to a queue manager, but the MQCD channel definition structure is not specified correctly. One of the following applies:

- `ClientConnOffset` is not zero and `ClientConnPtr` is not zero and not the null pointer.
- `ClientConnPtr` is not a valid pointer.
- `ClientConnPtr` or `ClientConnOffset` points to storage that is not accessible.

It also occurs in Java applications when a client channel definition table (CCDT) is specified to determine the name of the channel, but the table itself cannot be found.

Completion code

MQCC_FAILED

Programmer response

Ensure that at least one of `ClientConnOffset` and `ClientConnPtr` is zero. Ensure that the field used points to accessible storage. Ensure that the URL of the client channel definition table is correct.

2279 (08E7) (RC2279): MQRC_CHANNEL_STOPPED_BY_USER

Explanation

This condition is detected when the channel has been stopped by an operator. The reason qualifier identifies the reasons for stopping.

Completion code

MQCC_WARNING

Programmer response

None. This reason code is only used to identify the corresponding event message "Channel Stopped By User" on page 170.

Multi 2280 (08E8) (RC2280): MQRC_HCONFIG_ERROR

Explanation

The configuration handle `Hconfig` specified on the MQXEP call or MQZEP call is not valid. The MQXEP call is issued by an API exit function; the MQZEP call is issued by an installable service.

Completion code

MQCC_FAILED

Programmer response

Specify the configuration handle that was provided by the queue manager:

- On the MQXEP call, use the handle passed in the `Hconfig` field of the MQAXP structure.
- On the MQZEP call, use the handle passed to the installable service's configuration function on the component initialization call. For more information about installable services, see Installable services and components for AIX, Linux, and Windows .

Multi 2281 (08E9) (RC2281): MQRC_FUNCTION_ERROR

Explanation

An MQXEP or MQZEP call was issued, but the function identifier `Function` specified on the call is not valid, or not supported by the installable service being configured.

Completion code

MQCC_FAILED

Programmer response

Do the following:

- For the MQXEP call, specify one of the MQXF_* values.
- For the MQZEP call, specify an MQZID_* value that is valid for the installable service being configured. See MQZEP to determine which values are valid.

2282 (08EA) (RC2282): MQRC_CHANNEL_STARTED

Explanation

One of the following has occurred:

- An operator has issued a Start Channel command.
- An instance of a channel has been successfully established. This condition is detected when Initial Data negotiation is complete and resynchronization has been performed where necessary such that message transfer can proceed.

Completion code

MQCC_WARNING

Programmer response

None. This reason code is only used to identify the corresponding event message "Channel Started" on page 166.

2283 (08EB) (RC2283): MQRC_CHANNEL_STOPPED

Explanation

This condition is detected when the channel has been stopped. The reason qualifier identifies the reasons for stopping.

Completion code

MQCC_WARNING

Programmer response

None. This reason code is only used to identify the corresponding event message "Channel Stopped" on page 167.

2284 (08EC) (RC2284): MQRC_CHANNEL_CONV_ERROR

Explanation

This condition is detected when a channel is unable to do data conversion and the MQGET call to get a message from the transmission queue resulted in a data conversion error. The conversion reason code identifies the reason for the failure.

Completion code

MQCC_WARNING

Programmer response

None. This reason code is only used to identify the corresponding event message “Channel Conversion Error” on page 156.

Multi 2285 (08ED) (RC2285): MQRC_SERVICE_NOT_AVAILABLE

Explanation

This reason should be returned by an installable service component when the requested action cannot be performed because the required underlying service is not available.

Completion code

MQCC_FAILED

Programmer response

Make the underlying service available.

Multi 2286 (08EE) (RC2286): MQRC_INITIALIZATION_FAILED

Explanation

This reason should be returned by an installable service component when the component is unable to complete initialization successfully.

Completion code

MQCC_FAILED

Programmer response

Correct the error and retry the operation.

Multi 2287 (08EF) (RC2287): MQRC_TERMINATION_FAILED

Explanation

This reason should be returned by an installable service component when the component is unable to complete termination successfully.

Completion code

MQCC_FAILED

Programmer response

Correct the error and retry the operation.

Multi 2288 (08F0) (RC2288): MQRC_UNKNOWN_Q_NAME

Explanation

This reason should be returned by the MQZ_LOOKUP_NAME installable service component when the name specified for the **QName** parameter is not recognized.

Completion code

MQCC_FAILED

Programmer response

None. See Installable services and components for AIX, Linux, and Windows for more information about installable services.

Multi 2289 (08F1) (RC2289): MQRC_SERVICE_ERROR

Explanation

This reason should be returned by an installable service component when the component encounters an unexpected error.

Completion code

MQCC_FAILED

Programmer response

Correct the error and retry the operation.

Multi 2290 (08F2) (RC2290): MQRC_Q_ALREADY_EXISTS

Explanation

This reason should be returned by the MQZ_INSERT_NAME installable service component when the queue specified by the **QName** parameter is already defined to the name service.

Completion code

MQCC_FAILED

Programmer response

None. See Installable services and components for AIX, Linux, and Windows for more information about installable services.

Multi 2291 (08F3) (RC2291): MQRC_USER_ID_NOT_AVAILABLE

Explanation

This reason should be returned by the MQZ_FIND_USERID installable service component when the user ID cannot be determined.

Completion code

MQCC_FAILED

Programmer response

None. See Installable services and components for AIX, Linux, and Windows for more information about installable services.

Multi 2292 (08F4) (RC2292): MQRC_UNKNOWN_ENTITY

Explanation

This reason should be returned by the authority installable service component when the name specified by the **EntityName** parameter is not recognized.

Completion code

MQCC_FAILED

Programmer response

Ensure that the entity is defined.

Multi 2294 (08F6) (RC2294): MQRC_UNKNOWN_REF_OBJECT

Explanation

This reason should be returned by the MQZ_COPY_ALL_AUTHORITY installable service component when the name specified by the **RefObjectName** parameter is not recognized.

Completion code

MQCC_FAILED

Programmer response

Ensure that the reference object is defined. See Installable services and components for AIX, Linux, and Windows for more information about installable services.

2295 (08F7) (RC2295): MQRC_CHANNEL_ACTIVATED

Explanation

This condition is detected when a channel that has been waiting to become active, and for which a Channel Not Activated event has been generated, is now able to become active because an active slot has been released by another channel.

This event is not generated for a channel that is able to become active without waiting for an active slot to be released.

Completion code

MQCC_WARNING

Programmer response

None. This reason code is only used to identify the corresponding event message "Channel Activated" on page 150.

2296 (08F8) (RC2296): MQRC_CHANNEL_NOT_ACTIVATED

Explanation

This condition is detected when a channel is required to become active, either because it is starting or because it is about to make another attempt to establish connection with its partner. However, it is unable to do so because the limit on the number of active channels has been reached.

- z/OS On z/OS, the maximum number of active channels is given by the ACTCHL queue manager attribute.
- Multi In other environments, the maximum number of active channels is given by the MaxActiveChannels parameter in the `qm.ini` file.

The channel waits until it is able to take over an active slot released when another channel ceases to be active. At that time a Channel Activated event is generated.

Completion code

MQCC_WARNING

Programmer response

None. This reason code is only used to identify the corresponding event message "Channel Not Activated" on page 158.

2297 (08F9) (RC2297): MQRC_UOW_CANCELED

Explanation

An MQI call was issued, but the unit of work (TM/MP transaction) being used for the MQ operation had been canceled. This may have been done by TM/MP itself (for example, due to the transaction running for too long, or exceeding audit trail sizes), or by the application program issuing an ABORT_TRANSACTION. All updates performed to resources owned by the queue manager are backed out.

Completion code

MQCC_FAILED

Programmer response

Refer to the operating system's *Transaction Management Operations Guide* to determine how the Transaction Manager can be tuned to avoid the problem of system limits being exceeded.

2298 (08FA) (RC2298): MQRC_FUNCTION_NOT_SUPPORTED

Explanation

The function requested is not available in the current environment.

Completion code

MQCC_FAILED

Programmer response

Remove the call from the application.

z/OS If you get this reason code with CICS group connect, check that the queue manager attribute **GROUPUR** is enabled.

2299 (08FB) (RC2299): MQRC_SELECTOR_TYPE_ERROR

Explanation

The **Selector** parameter has the wrong data type; it must be of type Long.

Completion code

MQCC_FAILED

Programmer response

Declare the **Selector** parameter as Long.

2300 (08FC) (RC2300): MQRC_COMMAND_TYPE_ERROR

Explanation

The mqExecute call was issued, but the value of the MQIASY_TYPE data item in the administration bag is not MQCFT_COMMAND.

Completion code

MQCC_FAILED

Programmer response

Ensure that the MQIASY_TYPE data item in the administration bag has the value MQCFT_COMMAND.

2301 (08FD) (RC2301): MQRC_MULTIPLE_INSTANCE_ERROR

Explanation

The **Selector** parameter specifies a system selector (one of the MQIASY_* values), but the value of the **ItemIndex** parameter is not MQIND_NONE. Only one instance of each system selector can exist in the bag.

Completion code

MQCC_FAILED

Programmer response

Specify MQIND_NONE for the **ItemIndex** parameter.

2302 (08FE) (RC2302): MQRC_SYSTEM_ITEM_NOT_ALTERABLE

Explanation

A call was issued to modify the value of a system data item in a bag (a data item with one of the MQIASY_* selectors), but the call failed because the data item is one that cannot be altered by the application.

Completion code

MQCC_FAILED

Programmer response

Specify the selector of a user-defined data item, or remove the call.

2303 (08FF) (RC2303): MQRC_BAG_CONVERSION_ERROR

Explanation

The mqBufferToBag or mqGetBag call was issued, but the data in the buffer or message could not be converted into a bag. This occurs when the data to be converted is not valid PCF.

Completion code

MQCC_FAILED

Programmer response

Check the logic of the application that created the buffer or message to ensure that the buffer or message contains valid PCF.

If the message contains PCF that is not valid, the message cannot be retrieved using the mqGetBag call:

- If one of the MQGMO_BROWSE_* options was specified, the message remains on the queue and can be retrieved using the MQGET call.
- In other cases, the message has already been removed from the queue and discarded. If the message was retrieved within a unit of work, the unit of work can be backed out and the message retrieved using the MQGET call.

2304 (0900) (RC2304): MQRC_SELECTOR_OUT_OF_RANGE

Explanation

The **Selector** parameter has a value that is outside the valid range for the call. If the bag was created with the MQCBO_CHECK_SELECTORS option:

- For the mqAddInteger call, the value must be within the range MQIA_FIRST through MQIA_LAST.
- For the mqAddString call, the value must be within the range MQCA_FIRST through MQCA_LAST.

If the bag was not created with the MQCBO_CHECK_SELECTORS option, the value must be zero or greater.

Completion code

MQCC_FAILED

Programmer response

Specify a valid value.

2305 (0901) (RC2305): MQRC_SELECTOR_NOT_UNIQUE

Explanation

The **ItemIndex** parameter has the value MQIND_NONE, but the bag contains more than one data item with the selector value specified by the **Selector** parameter. MQIND_NONE requires that the bag contain only one occurrence of the specified selector.

This reason code also occurs on the mqExecute call when the administration bag contains two or more occurrences of a selector for a required parameter that permits only one occurrence.

Completion code

MQCC_FAILED

Programmer response

Check the logic of the application that created the bag. If correct, specify for *ItemIndex* a value that is zero or greater, and add application logic to process all of the occurrences of the selector in the bag.

Review the description of the administration command being issued, and ensure that all required parameters are defined correctly in the bag.

2306 (0902) (RC2306): MQRC_INDEX_NOT_PRESENT

Explanation

The specified index is not present:

- For a bag, this means that the bag contains one or more data items that have the selector value specified by the **Selector** parameter, but none of them has the index value specified by the **ItemIndex** parameter. The data item identified by the **Selector** and **ItemIndex** parameters must exist in the bag.
- For a namelist, this means that the index parameter value is too large, and outside the range of valid values.

Completion code

MQCC_FAILED

Programmer response

Specify the index of a data item that does exist in the bag or namelist. Use the mqCountItems call to determine the number of data items with the specified selector that exist in the bag, or the nameCount method to determine the number of names in the namelist.

2307 (0903) (RC2307): MQRC_STRING_ERROR

Explanation

The **String** parameter is not valid. Either the parameter pointer is not valid, or it points to read-only storage. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)

Completion code

MQCC_FAILED

Programmer response

Correct the parameter.

2308 (0904) (RC2308): MQRC_ENCODING_NOT_SUPPORTED

Explanation

The Encoding field in the message descriptor MQMD contains a value that is not supported:

- For the mqPutBag call, the field in error resides in the **MsgDesc** parameter of the call.
- For the mqGetBag call, the field in error resides in:
  - The **MsgDesc** parameter of the call if the MQGMO_CONVERT option was specified.
  - The message descriptor of the message about to be retrieved if MQGMO_CONVERT was not specified.

Completion code

MQCC_FAILED

Programmer response

The value must be MQENC_NATIVE.

If the value of the Encoding field in the message is not valid, the message cannot be retrieved using the mqGetBag call:

- If one of the MQGMO_BROWSE_* options was specified, the message remains on the queue and can be retrieved using the MQGET call.
- In other cases, the message has already been removed from the queue and discarded. If the message was retrieved within a unit of work, the unit of work can be backed out and the message retrieved using the MQGET call.

2309 (0905) (RC2309): MQRC_SELECTOR_NOT_PRESENT

Explanation

The **Selector** parameter specifies a selector that does not exist in the bag.

Completion code

MQCC_FAILED

Programmer response

Specify a selector that does exist in the bag.

2310 (0906) (RC2310): MQRC_OUT_SELECTOR_ERROR

Explanation

The **OutSelector** parameter is not valid. Either the parameter pointer is not valid, or it points to read-only storage. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)

Completion code

MQCC_FAILED

Programmer response

Correct the parameter.

2311 (0907) (RC2311): MQRC_STRING_TRUNCATED

Explanation

The string returned by the call is too long to fit in the buffer provided. The string has been truncated to fit in the buffer.

Completion code

MQCC_FAILED

Programmer response

If the entire string is required, provide a larger buffer. On the mqInquireString call, the **StringLength** parameter is set by the call to indicate the size of the buffer required to accommodate the string without truncation.

2312 (0908) (RC2312): MQRC_SELECTOR_WRONG_TYPE

Explanation

A data item with the specified selector exists in the bag, but has a data type that conflicts with the data type implied by the call being used. For example, the data item might have an integer data type, but the call being used might be mqSetString, which implies a character data type.

This reason code also occurs on the mqBagToBuffer, mqExecute, and mqPutBag calls when mqAddString or mqSetString was used to add the MQIACF_INQUIRY data item to the bag.

Completion code

MQCC_FAILED

Programmer response

For the mqSetInteger and mqSetString calls, specify MQIND_ALL for the **ItemIndex** parameter to delete from the bag all existing occurrences of the specified selector before creating the new occurrence with the required data type.

For the mqInquireBag, mqInquireInteger, and mqInquireString calls, use the mqInquireItemInfo call to determine the data type of the item with the specified selector, and then use the appropriate call to determine the value of the data item.

For the mqBagToBuffer, mqExecute, and mqPutBag calls, ensure that the MQIACF_INQUIRY data item is added to the bag using the mqAddInteger or mqSetInteger calls.

2313 (0909) (RC2313): MQRC_INCONSISTENT_ITEM_TYPE

Explanation

The mqAddInteger or mqAddString call was issued to add another occurrence of the specified selector to the bag, but the data type of this occurrence differed from the data type of the first occurrence.

This reason can also occur on the mqBufferToBag and mqGetBag calls, where it indicates that the PCF in the buffer or message contains a selector that occurs more than once but with inconsistent data types.

Completion code

MQCC_FAILED

Programmer response

For the mqAddInteger and mqAddString calls, use the call appropriate to the data type of the first occurrence of that selector in the bag.

For the mqBufferToBag and mqGetBag calls, check the logic of the application that created the buffer or sent the message to ensure that multiple-occurrence selectors occur with only one data type. A message that contains a mixture of data types for a selector cannot be retrieved using the mqGetBag call:

- If one of the MQGMO_BROWSE_* options was specified, the message remains on the queue and can be retrieved using the MQGET call.
- In other cases, the message has already been removed from the queue and discarded. If the message was retrieved within a unit of work, the unit of work can be backed out and the message retrieved using the MQGET call.

2314 (090A) (RC2314): MQRC_INDEX_ERROR

Explanation

An index parameter to a call or method has a value that is not valid. The value must be zero or greater. For bag calls, certain MQIND_* values can also be specified:

- For the mqDeleteItem, mqSetInteger and mqSetString calls, MQIND_ALL and MQIND_NONE are valid.
- For the mqInquireBag, mqInquireInteger, mqInquireString, and mqInquireItemInfo calls, MQIND_NONE is valid.

Completion code

MQCC_FAILED

Programmer response

Specify a valid value.

2315 (090B) (RC2315): MQRC_SYSTEM_BAG_NOT_ALTERABLE

Explanation

A call was issued to add a data item to a bag, modify the value of an existing data item in a bag, or retrieve a message into a bag, but the call failed because the bag is one that had been created by the system as a result of a previous mqExecute call. System bags cannot be modified by the application.

Completion code

MQCC_FAILED

Programmer response

Specify the handle of a bag created by the application, or remove the call.

2316 (090C) (RC2316): MQRC_ITEM_COUNT_ERROR

Explanation

The mqTruncateBag call was issued, but the **ItemCount** parameter specifies a value that is not valid. The value is either less than zero, or greater than the number of user-defined data items in the bag.

This reason also occurs on the mqCountItems call if the parameter pointer is not valid, or points to read-only storage. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)

Completion code

MQCC_FAILED

Programmer response

Specify a valid value. Use the mqCountItems call to determine the number of user-defined data items in the bag.

2317 (090D) (RC2317): MQRC_FORMAT_NOT_SUPPORTED

Explanation

The *Format* field in the message descriptor MQMD contains a value that is not supported:

- In an administration message, the format value must be one of the following: MQFMT_ADMIN, MQFMT_EVENT, MQFMT_PCF. For the mqPutBag call, the field in error resides in the **MsgDesc** parameter of the call. For the mqGetBag call, the field in error resides in the message descriptor of the message about to be retrieved.
- z/OS On z/OS, the message was put to the command input queue with a format value of MQFMT_ADMIN, but the version of IBM MQ being used does not support that format for commands.

Completion code

MQCC_FAILED

Programmer response

If the error occurred when putting a message, correct the format value.

If the error occurred when getting a message, the message cannot be retrieved using the mqGetBag call:

- If one of the MQGMO_BROWSE_* options was specified, the message remains on the queue and can be retrieved using the MQGET call.
- In other cases, the message has already been removed from the queue and discarded. If the message was retrieved within a unit of work, the unit of work can be backed out and the message retrieved using the MQGET call.

2318 (090E) (RC2318): MQRC_SELECTOR_NOT_SUPPORTED

Explanation

The **Selector** parameter specifies a value that is a system selector (a value that is negative), but the system selector is not one that is supported by the call.

Completion code

MQCC_FAILED

Programmer response

Specify a selector value that is supported.

2319 (090F) (RC2319): MQRC_ITEM_VALUE_ERROR

Explanation

The mqInquireBag or mqInquireInteger call was issued, but the **ItemValue** parameter is not valid. Either the parameter pointer is not valid, or it points to read-only storage. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)

Completion code

MQCC_FAILED

Programmer response

Correct the parameter.

2320 (0910) (RC2320): MQRC_HBAG_ERROR

Explanation

A call was issued that has a parameter that is a bag handle, but the handle is not valid. For output parameters, this reason also occurs if the parameter pointer is not valid, or points to read-only storage. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)

Completion code

MQCC_FAILED

Programmer response

Correct the parameter.

2321 (0911) (RC2321): MQRC_PARAMETER_MISSING

Explanation

An administration message requires a parameter that is not present in the administration bag. This reason code occurs only for bags created with the MQCBO_ADMIN_BAG or MQCBO_REORDER_AS_REQUIRED options.

Completion code

MQCC_FAILED

Programmer response

Review the description of the administration command being issued, and ensure that all required parameters are present in the bag.

2322 (0912) (RC2322): MQRC_CMD_SERVER_NOT_AVAILABLE

Explanation

The command server that processes administration commands is not available.

Completion code

MQCC_FAILED

Programmer response

Start the command server.

2323 (0913) (RC2323): MQRC_STRING_LENGTH_ERROR

Explanation

The **StringLength** parameter is not valid. Either the parameter pointer is not valid, or it points to read-only storage. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)

Completion code

MQCC_FAILED

Programmer response

Correct the parameter.

2324 (0914) (RC2324): MQRC_INQUIRY_COMMAND_ERROR

Explanation

The mqAddInquiry call was used previously to add attribute selectors to the bag, but the command code to be used for the mqBagToBuffer, mqExecute, or mqPutBag call is not recognized. As a result, the correct PCF message cannot be generated.

Completion code

MQCC_FAILED

Programmer response

Remove the mqAddInquiry calls and use instead the mqAddInteger call with the appropriate MQIACF_*_ATTRS or MQIACH_*_ATTRS selectors.

2325 (0915) (RC2325): MQRC_NESTED_BAG_NOT_SUPPORTED

Explanation

A bag that is input to the call contains nested bags. Nested bags are supported only for bags that are output from the call.

Completion code

MQCC_FAILED

Programmer response

Use a different bag as input to the call.

2326 (0916) (RC2326): MQRC_BAG_WRONG_TYPE

Explanation

The **Bag** parameter specifies the handle of a bag that has the wrong type for the call. The bag must be an administration bag, that is, it must be created with the MQCBO_ADMIN_BAG option specified on the mqCreateBag call.

Completion code

MQCC_FAILED

Programmer response

Specify the MQCBO_ADMIN_BAG option when the bag is created.

2327 (0917) (RC2327): MQRC_ITEM_TYPE_ERROR

Explanation

The mqInquireItemInfo call was issued, but the **ItemType** parameter is not valid. Either the parameter pointer is not valid, or it points to read-only storage. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)

Completion code

MQCC_FAILED

Programmer response

Correct the parameter.

2328 (0918) (RC2328): MQRC_SYSTEM_BAG_NOT_DELETABLE

Explanation

An mqDeleteBag call was issued to delete a bag, but the call failed because the bag is one that had been created by the system as a result of a previous mqExecute call. System bags cannot be deleted by the application.

Completion code

MQCC_FAILED

Programmer response

Specify the handle of a bag created by the application, or remove the call.

2329 (0919) (RC2329): MQRC_SYSTEM_ITEM_NOT_DELETABLE

Explanation

A call was issued to delete a system data item from a bag (a data item with one of the MQIASY_* selectors), but the call failed because the data item is one that cannot be deleted by the application.

Completion code

MQCC_FAILED

Programmer response

Specify the selector of a user-defined data item, or remove the call.

2330 (091A) (RC2330): MQRC_CODED_CHAR_SET_ID_ERROR

Explanation

The **CodedCharSetId** parameter is not valid. Either the parameter pointer is not valid, or it points to read-only storage. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)

Completion code

MQCC_FAILED

Programmer response

Correct the parameter.

2331 (091B) (RC2331): MQRC_MSG_TOKEN_ERROR

Explanation

An MQGET call was issued to retrieve a message using the message token as a selection criterion, but the options specified are not valid, because MQMO_MATCH_MSG_TOKEN was specified with either MQGMO_WAIT or MQGMO_SET_SIGNAL.

An Async Consumer was registered to retrieve a message using the message token as a selection criterion, but when the delivery of messages for this consumer was started no message matching the message token was available for delivery to the consumer. As a result the consumer is suspended.

Completion code

MQCC_FAILED

Programmer response

If this reason code is returned from an MQGET call, either remove the MQMO_MATCH_MSG_TOKEN match option, or remove the MQGMO_WAIT, or MQGMO_SET_SIGNAL option which was specified.

If this reason code is returned to an Async Consume Event Handler, then the consumer has been suspended and no further messages will be delivered to the consumer. The consumer should be de-registered or modified to select a different message using the MQCB call.

z/OS 2332 (091C) (RC2332): MQRC_MISSING_WIH

Explanation

An MQPUT or MQPUT1 call was issued to put a message on a queue with an **IndexType** attribute that had the value MQIT_MSG_TOKEN, but the `Format` field in the MQMD was not MQFMT_WORK_INFO_HEADER. This error occurs only when the message arrives at the destination queue manager.

Completion code

MQCC_FAILED

Programmer response

Modify the application to ensure that it places an MQWIH structure at the start of the message data, and sets the `Format` field in the MQMD to MQFMT_WORK_INFO_HEADER. Alternatively, change the **ApplType** attribute of the process definition used by the destination queue to be MQAT_WLM, and specify the required service name and service step name in its **EnvData** attribute.

2333 (091D) (RC2333): MQRC_WIH_ERROR

Explanation

An MQPUT or MQPUT1 call was issued, but the message data contains an MQWIH structure that is not valid. Possible errors include the following:

- The *StrucId* field is not MQWIH_STRUC_ID.
- The *Version* field is not MQWIH_VERSION_1.
- The *StrucLength* field is not MQWIH_LENGTH_1.
- The *CodedCharSetId* field is zero, or a negative value that is not valid.
- The **BufferLength** parameter of the call has a value that is too small to accommodate the structure (the structure extends beyond the end of the message).

z/OS On z/OS, this error also occurs when the **IndexType** attribute of the queue is MQIT_MSG_TOKEN, but the message data does not begin with an MQWIH structure.

Completion code

MQCC_FAILED

Programmer response

Check that the fields in the structure are set correctly. Ensure that the application sets the `CodedCharSetId` field to a valid value (note: MQCCSI_DEFAULT, MQCCSI_EMBEDDED, MQCCSI_Q_MGR, and MQCCSI_UNDEFINED are not valid in this field).

z/OS On z/OS, if the queue has an `IndexType` of MQIT_MSG_TOKEN, ensure that the message data begins with an MQWIH structure.

2334 (091E) (RC2334): MQRC_RFH_ERROR

Explanation

An MQPUT or MQPUT1 call was issued, but the message data contains an MQRFH or MQRFH2 structure that is not valid. Possible errors include the following:

- The `StrucId` field is not MQRFH_STRUC_ID.
- The `Version` field is not MQRFH_VERSION_1 (MQRFH), or MQRFH_VERSION_2 (MQRFH2).
- The `StrucLength` field specifies a value that is too small to include the structure plus the variable-length data at the end of the structure.
- The `CodedCharSetId` field is zero, or a negative value that is not valid.
- The **`BufferLength`** parameter of the call has a value that is too small to accommodate the structure (the structure extends beyond the end of the message).

Completion code

MQCC_FAILED

Programmer response

Check that the fields in the structure are set correctly. Ensure that the application sets the `CodedCharSetId` field to a valid value (note: MQCCSI_DEFAULT, MQCCSI_EMBEDDED, MQCCSI_Q_MGR, and MQCCSI_UNDEFINED are not valid in this field).

2335 (091F) (RC2335): MQRC_RFH_STRING_ERROR

Explanation

The contents of the `NameValueString` field in the MQRFH structure are not valid. `NameValueString` must adhere to the following rules:

- The string must consist of zero or more name-value pairs separated from each other by one or more blanks; the blanks are not significant.
- If a name or value contains blanks that are significant, the name or value must be enclosed in double quotation marks.
- If a name or value itself contains one or more double quotation marks, the name or value must be enclosed in double quotation marks, and each embedded double quotation mark must be doubled.
- A name or value can contain any characters other than the null, which acts as a delimiter. The null and characters following it, up to the defined length of `NameValueString`, are ignored.

The following is a valid NameValueString:

```
Famous_Words "The program displayed ""Hello World"""
```

Completion code

MQCC_FAILED

Programmer response

Modify the application that generated the message to ensure that it places in the NameValueString field data that adheres to the rules. Check that the StrucLength field is set to the correct value.

2336 (0920) (RC2336): MQRC_RFH_COMMAND_ERROR

Explanation

The message contains an MQRFH structure, but the command name contained in the NameValueString field is not valid.

Completion code

MQCC_FAILED

Programmer response

Modify the application that generated the message to ensure that it places in the NameValueString field a command name that is valid.

2337 (0921) (RC2337): MQRC_RFH_PARM_ERROR

Explanation

The message contains an MQRFH structure, but a parameter name contained in the NameValueString field is not valid for the command specified.

Completion code

MQCC_FAILED

Programmer response

Modify the application that generated the message to ensure that it places in the NameValueString field only parameters that are valid for the specified command.

2338 (0922) (RC2338): MQRC_RFH_DUPLICATE_PARM

Explanation

The message contains an MQRFH structure, but a parameter occurs more than once in the NameValueString field when only one occurrence is valid for the specified command.

Completion code

MQCC_FAILED

Programmer response

Modify the application that generated the message to ensure that it places in the `NameValueString` field only one occurrence of the parameter.

2339 (0923) (RC2339): MQRC_RFH_PARM_MISSING

Explanation

The message contains an MQRFH structure, but the command specified in the `NameValueString` field requires a parameter that is not present.

Completion code

MQCC_FAILED

Programmer response

Modify the application that generated the message to ensure that it places in the `NameValueString` field all parameters that are required for the specified command.

z/OS 2340 (0924) (RC2340): MQRC_CHAR_CONVERSION_ERROR

Explanation

This reason code is returned by the Java MQQueueManager constructor when a required character-set conversion is not available. The conversion required is between two nonUnicode character sets.

This reason code occurs in the following environment: IBM MQ classes for Java on z/OS.

Completion code

MQCC_FAILED

Programmer response

Ensure that the National Language Resources component of the z/OS Language Environment is installed, and that conversion between the IBM-1047 and ISO8859-1 character sets is available.

2341 (0925) (RC2341): MQRC_UCS2_CONVERSION_ERROR

Explanation

This reason code is returned by the Java MQQueueManager constructor when a required character set conversion is not available. The conversion required is between the UTF-16 Unicode character set and the character set of the queue manager which defaults to IBM-500 if no specific value is available.

Completion code

MQCC_FAILED

Programmer response

Ensure that the relevant Unicode conversion tables are available for the JVM.

z/OS For z/OS ensure that the Unicode conversion tables are available to the z/OS Language Environment. The conversion tables should be installed as part of the z/OS C/C++ optional feature. Refer to the *z/OS C/C++ Programming Guide* for more information about enabling UTF-16 conversions.

z/OS 2342 (0926) (RC2342): MQRC_DB2_NOT_AVAILABLE

Explanation

An MQOPEN, MQPUT1, or MQSET call, or a command, was issued to access a shared queue, but it failed because the queue manager is not connected to a Db2 subsystem. As a result, the queue manager is unable to access the object definition relating to the shared queue.

This reason code occurs only on z/OS.

Completion code

MQCC_FAILED

Programmer response

Configure the Db2 subsystem so that the queue manager can connect to it.

z/OS 2343 (0927) (RC2343): MQRC_OBJECT_NOT_UNIQUE

Explanation

An MQOPEN or MQPUT1 call, or a command, was issued to access a queue, but the call failed because the queue specified cannot be resolved unambiguously. There exists a shared queue with the specified name, and a nonshared queue with the same name.

This reason code occurs only on z/OS.

Completion code

MQCC_FAILED

Programmer response

One of the queues must be deleted. If the queue to be deleted contains messages, use the MQSC command MOVE QLOCAL to move the messages to a different queue, and then use the command DELETE QLOCAL to delete the queue.

z/OS 2344 (0928) (RC2344): MQRC_CONN_TAG_NOT_RELEASED

Explanation

An MQDISC call was issued when there was a unit of work outstanding for the connection handle. For CICS, IMS, and RRS connections, the MQDISC call does not commit or back out the unit of work. As a result, the connection tag associated with the unit of work is not yet available for reuse. The tag becomes available for reuse only when processing of the unit of work has been completed.

This reason code occurs only on z/OS.

Completion code

MQCC_WARNING

Programmer response

Do not try to reuse the connection tag immediately. If the MQCONNX call is issued with the same connection tag, and that tag is still in use, the call fails with reason code MQRC_CONN_TAG_IN_USE.

z/OS 2345 (0929) (RC2345): MQRC_CF_NOT_AVAILABLE

Explanation

An MQI call was issued to access a shared queue, but the call failed either because connectivity was lost to the coupling facility (CF) where the CF structure specified in the queue definition was allocated, or because allocation of the CF structure failed because there is no suitable CF to hold the structure, based on the preference list in the active CFRM policy.

This reason code occurs only on z/OS.

Completion code

MQCC_FAILED

Programmer response

If connectivity was lost to the CF where the structure was allocated, and the queue manager has been configured to tolerate the failure and rebuild the structure, no action should be necessary. Otherwise, make available a coupling facility with one of the names specified in the CFRM policy, or modify the CFRM policy to specify the names of coupling facilities that are available.

z/OS 2346 (092A) (RC2346): MQRC_CF_STRUC_IN_USE

Explanation

An MQI call or command was issued to operate on a shared queue, but the call failed because the coupling-facility structure specified in the queue definition is unavailable. The coupling-facility structure can be unavailable because a structure dump is in progress, or new connectors to the structure are currently inhibited, or an existing connector to the structure failed or disconnected abnormally and clean-up is not yet complete.

This reason code occurs only on z/OS.

Completion code

MQCC_FAILED

Programmer response

Typically, this is a temporary problem: wait for a while then retry the operation.

If the problem does not resolve itself, then connectivity problems experienced during the recovery of structures in the coupling facility could have occurred. In this case, restart the queue manager which reported the error. Resolve all the connectivity problems concerning the coupling facility before restarting the queue manager.

z/OS 2347 (092B) (RC2347): MQRC_CF_STRUC_LIST_HDR_IN_USE

Explanation

An MQGET, MQOPEN, MQPUT1, or MQSET call was issued to access a shared queue, but the call failed because the list header associated with the coupling-facility structure specified in the queue definition is temporarily unavailable. The list header is unavailable because it is undergoing recovery processing.

This reason code occurs only on z/OS.

Completion code

MQCC_FAILED

Programmer response

The problem is temporary; wait a short while and then retry the operation.

z/OS 2348 (092C) (RC2348): MQRC_CF_STRUC_AUTH_FAILED

Explanation

An MQOPEN or MQPUT1 call was issued to access a shared queue, but the call failed because the user is not authorized to access the coupling-facility structure specified in the queue definition.

This reason code occurs only on z/OS.

Completion code

MQCC_FAILED

Programmer response

Modify the security profile for the user identifier used by the application so that the application can access the coupling-facility structure specified in the queue definition.

z/OS 2349 (092D) (RC2349): MQRC_CF_STRUC_ERROR

Explanation

An MQOPEN or MQPUT1 call was issued to access a shared queue, but the call failed because the coupling-facility structure name specified in the queue definition is not defined in the CFRM data set, or is not the name of a list structure.

This reason code occurs only on z/OS.

Completion code

MQCC_FAILED

Programmer response

Modify the queue definition to specify the name of a coupling-facility list structure that is defined in the CFRM data set.

z/OS 2350 (092E) (RC2350): MQRC_CONN_TAG_NOT_USABLE

Explanation

An MQCONNX call was issued specifying one of the MQCNO_*_CONN_TAG_* options, but the call failed because the connection tag specified by ConnTag in MQCNO is being used by the queue manager for recovery processing, and this processing is delayed pending recovery of the coupling facility.

This reason code occurs only on z/OS.

Completion code

MQCC_FAILED

Programmer response

The problem is likely to persist. Consult the system programmer to ascertain the cause of the problem.

Windows z/OS 2351 (092F) (RC2351): MQRC_GLOBAL_UOW_CONFLICT

Explanation

An attempt was made to use inside a global unit of work a connection handle that is participating in another global unit of work. This can occur when an application passes connection handles between objects where the objects are involved in different DTC transactions. Because transaction completion is asynchronous, it is possible for this error to occur after the application has finalized the first object and committed its transaction.

This error does not occur for non-transactional MQI calls.

This reason code occurs only on Windows and z/OS.

Completion code

MQCC_FAILED

Programmer response

Check that the **MTS Transaction Support** attribute defined for the object's class is set correctly. If necessary, modify the application so that the connection handle is not used by objects participating in different units of work.

Windows z/OS 2352 (0930) (RC2352): MQRC_LOCAL_UOW_CONFLICT

Explanation

An attempt was made to use inside a global unit of work a connection handle that is participating in a queue manager coordinated local unit of work. This can occur when an application passes connection handles between objects where one object is involved in a DTC transaction and the other is not.

This error does not occur for non-transactional MQI calls.

This reason code occurs only on Windows and z/OS.

Completion code

MQCC_FAILED

Programmer response

Check that the **MTS Transaction Support** attribute defined for the object's class is set correctly. If necessary, modify the application so that the connection handle is not used by objects participating in different units of work.

Windows 2353 (0931) (RC2353): MQRC_HANDLE_IN_USE_FOR_UOW

Explanation

An attempt was made to use outside a unit of work a connection handle that is participating in a global unit of work.

This error can occur when an application passes connection handles between objects where one object is involved in a DTC transaction and the other is not. Because transaction completion is asynchronous, it is possible for this error to occur after the application has finalized the first object and committed its transaction.

This error can also occur when a single object that was created and associated with the transaction loses that association whilst the object is running. The association is lost when DTC terminates the transaction independently of MTS. This might be because the transaction timed out, or because DTC shut down.

This error does not occur for non-transactional MQI calls.

This reason code occurs only on Windows.

Completion code

MQCC_FAILED

Programmer response

Check that the MTS Transaction Support attribute defined for the object's class is set correctly. If necessary, modify the application so that objects executing within different units of work do not try to use the same connection handle.

Windows 2354 (0932) (RC2354): MQRC_UOW_ENLISTMENT_ERROR

Explanation

This reason code can occur for various reasons and occurs only on Windows.

The most likely reason is that an object created by a DTC transaction does not issue a transactional MQI call until after the DTC transaction timed out. (If the DTC transaction times out after a transactional MQI call has been issued, reason code MQRC_HANDLE_IN_USE_FOR_UOW is returned by the failing MQI call.)

Another cause of MQRC_UOW_ENLISTMENT_ERROR is incorrect installation.

Completion code

MQCC_FAILED

Programmer response

Windows On Windows, check the DTC Transaction timeout value. If necessary, verify the Windows installation order.

Windows 2355 (0933) (RC2355): MQRC_UOW_MIX_NOT_SUPPORTED

Explanation

This reason code occurs on Windows when you are running a version of the queue manager before IBM WebSphere MQ 5.2.

Windows On Windows, the following explanations might apply:

- The mixture of calls that is used by the application to perform operations within a unit of work is not supported. In particular, it is not possible to mix within the same process a local unit of work that is coordinated by the queue manager with a global unit of work that is coordinated by DTC (Distributed Transaction Coordinator).
- An application might cause this mixture to arise if some objects in a package are coordinated by DTC and others are not. It can also occur if transactional MQI calls from an MTS client are mixed with transactional MQI calls from a library package transactional MTS object.
- No problem arises if all transactional MQI calls originate from transactional MTS objects, or all transactional MQI calls originate from non-transactional MTS objects. But when a mixture of styles is used, the first style that is used fixes the style for the unit of work, and subsequent attempts to use the other style within the process fail with reason code MQRC_UOW_MIX_NOT_SUPPORTED.
- When an application is run twice, scheduling factors in the operating system mean that it is possible for the queue manager-coordinated transactional calls to fail in one run, and for the DTC-coordinated transactional calls to fail in the other run.

Completion code

MQCC_FAILED

Programmer response

Windows On Windows, check that the MTS Transaction Support attribute defined for the object's class is set correctly. If necessary, modify the application so that objects that run within different units of work do not try to use the same connection handle.

2356 (0934) (RC2356): MQRC_WXP_ERROR

Explanation

An MQXCLWLN call was issued from a cluster workload exit to obtain the address of the next record in the chain, but the workload exit parameter structure `ExitParms` is not valid, for one of the following reasons:

- The parameter pointer is not valid. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)
- The `StrucId` field is not MQWXP_STRUC_ID.
- The `Version` field is not MQWXP_VERSION_2.
- The `CacheContext` field does not contain the value passed to the exit by the queue manager.

Completion code

MQCC_FAILED

Programmer response

Ensure that the parameter specified for `ExitParms` is the MQWXP structure that was passed to the exit when the exit was invoked.

2357 (0935) (RC2357): MQRC_CURRENT_RECORD_ERROR

Explanation

An MQXCLWLN call was issued from a cluster workload exit to obtain the address of the next record in the chain, but the address specified by the **CurrentRecord** parameter is not the address of a valid record. `CurrentRecord` must be the address of a destination record (MQWDR), queue record (MQWQR), or cluster record (MQWCR) residing within the cluster cache.

Completion code

MQCC_FAILED

Programmer response

Ensure that the cluster workload exit passes the address of a valid record residing in the cluster cache.

2358 (0936) (RC2358): MQRC_NEXT_OFFSET_ERROR

Explanation

An MQXCLWLN call was issued from a cluster workload exit to obtain the address of the next record in the chain, but the offset specified by the **NextOffset** parameter is not valid. `NextOffset` must be the value of one of the following fields:

- `ChannelDefOffset` field in MQWDR
- `ClusterRecOffset` field in MQWDR
- `ClusterRecOffset` field in MQWQR
- `ClusterRecOffset` field in MQWCR

Completion code

MQCC_FAILED

Programmer response

Ensure that the value specified for the **NextOffset** parameter is the value of one of the fields listed.

2359 (0937) (RC2359): MQRC_NO_RECORD_AVAILABLE

Explanation

An MQXCLWLN call was issued from a cluster workload exit to obtain the address of the next record in the chain, but the current record is the last record in the chain.

Completion code

MQCC_FAILED

Programmer response

None.

z/OS 2360 (0938) (RC2360): MQRC_OBJECT_LEVEL_INCOMPATIBLE

Explanation

An MQOPEN or MQPUT1 call, or a command, was issued, but the definition of the object to be accessed is not compatible with the queue manager to which the application has connected. The object definition was created or modified by a different version of the queue manager.

If the object to be accessed is a queue, the incompatible object definition could be the object specified, or one of the object definitions used to resolve the specified object (for example, the base queue to which an alias queue resolves, or the transmission queue to which a remote queue or queue manager alias resolves).

This reason code occurs only on z/OS.

Completion code

MQCC_FAILED

Programmer response

The application must be run on a queue manager that is compatible with the object definition. .

2361 (0939) (RC2361): MQRC_NEXT_RECORD_ERROR

Explanation

An MQXCLWLN call was issued from a cluster workload exit to obtain the address of the next record in the chain, but the address specified for the **NextRecord** parameter is either null, not valid, or the address of read-only storage. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)

Completion code

MQCC_FAILED

Programmer response

Specify a valid address for the **NextRecord** parameter.

2362 (093A) (RC2362): MQRC_BACKOUT_THRESHOLD_REACHED

Explanation

This reason code occurs only in the Reason field in an MQDLH structure, or in the Feedback field in the MQMD of a report message.

A JMS ConnectionConsumer found a message that exceeds the queue's backout threshold. The queue does not have a backout requeue queue defined, so the message was processed as specified by the disposition options in the Report field in the MQMD of the message.

On queue managers that do not support the **BackoutThreshold** and **BackoutRequeueQName** queue attributes, JMS ConnectionConsumer uses a value of 20 for the backout threshold. When the BackoutCount of a message reaches this threshold, the message is processed as specified by the disposition options.

If the Report field specifies one of the MQRO_EXCEPTION_* options, this reason code appears in the Feedback field of the report message. If the Report field specifies MQRO_DEAD_LETTER_Q, or the

disposition report options remain at the default, this reason code appears in the Reason field of the MQDLH.

Completion code

None

Programmer response

Investigate the cause of the backout count being greater than the threshold. To correct this, define the backout queue for the queue concerned.

2363 (093B) (RC2363): MQRC_MSG_NOT_MATCHED

Explanation

This reason code occurs only in the Reason field in an MQDLH structure, or in the Feedback field in the MQMD of a report message.

While performing Point-to-Point messaging, JMS encountered a message matching none of the selectors of ConnectionConsumers monitoring the queue. To maintain performance, the message was processed as specified by the disposition options in the Report field in the MQMD of the message.

If the Report field specifies one of the MQRO_EXCEPTION_* options, this reason code appears in the Feedback field of the report message. If the Report field specifies MQRO_DEAD_LETTER_Q, or the disposition report options remain at the default, this reason code appears in the Reason field of the MQDLH.

Completion code

None

Programmer response

To correct this, ensure that the ConnectionConsumers monitoring the queue provide a complete set of selectors. Alternatively, set the QueueConnectionFactory to retain messages.

2364 (093C) (RC2364): MQRC_JMS_FORMAT_ERROR

Explanation

This reason code is generated by JMS applications that use one of the following:

- ConnectionConsumers
- Activation specifications
- WebSphere Application Server listener ports

and connect to an IBM MQ queue manager using IBM MQ messaging provider migration mode.

When the IBM MQ classes for JMS encounter a message that cannot be parsed (for example, the message contains an invalid RFH2 header) the message is processed as specified by the disposition options in the Report field in the MQMD of the message.

If the Report field specifies one of the MQRO_EXCEPTION_* options, this reason code appears in the Feedback field of the report message. If the Report field specifies MQRO_DEAD_LETTER_Q, or the disposition report options remain at the default, this reason code appears in the Reason field of the MQDLH.

Completion code

None

Programmer response

Investigate the origin of the message.

z/OS 2365 (093D) (RC2365): MQRC_SEGMENTS_NOT_SUPPORTED

Explanation

An MQPUT call was issued to put a segment of a logical message, but the queue on which the message is to be placed has an `IndexType` of MQIT_GROUP_ID. Message segments cannot be placed on queues with this index type.

This reason code occurs only on z/OS.

Completion code

MQCC_FAILED

Programmer response

Modify the application to put messages that are not segments; ensure that the MQMF_SEGMENT and MQMF_LAST_SEGMENT flags in the `MsgFlags` field in MQMD are not set, and that the `Offset` is zero. Alternatively, change the index type of the queue.

z/OS 2366 (093E) (RC2366): MQRC_WRONG_CF_LEVEL

Explanation

An MQOPEN, MQPUT, or MQPUT1 call was issued specifying a shared queue, but the queue requires a coupling-facility structure with a different level of capability.

For example an MQPUT or MQPUT1 call was issued with a message that was bigger than 63 KB in length, and the shared queue was using an IBM MQ CFSTRUCT with a CFLEVEL of less than 4.

This reason code occurs only on z/OS.

Completion code

MQCC_FAILED

Programmer response

Ensure that the coupling-facility structure used for the queue is at the level required to support the capabilities that the queue provides.

You can use the DISPLAY CFSTRUCT command to display the level, and ALTER CFSTRUCT() CFLEVEL() command to modify the level; see The MQSC commands.

2367 (093F) (RC2367): MQRC_CONFIG_CREATE_OBJECT

Explanation

This condition is detected when an object is created.

Completion code

MQCC_WARNING

Programmer response

None. This reason code is only used to identify the corresponding event message “Create object” on page 179.

2368 (0940) (RC2368): MQRC_CONFIG_CHANGE_OBJECT

Explanation

This condition is detected when an object is changed.

Completion code

MQCC_WARNING

Programmer response

None. This reason code is only used to identify the corresponding event message “Change object” on page 146.

2369 (0941) (RC2369): MQRC_CONFIG_DELETE_OBJECT

Explanation

This condition is detected when an object is deleted.

Completion code

MQCC_WARNING

Programmer response

None. This reason code is only used to identify the corresponding event message “Delete object” on page 190.

2370 (0942) (RC2370): MQRC_CONFIG_REFRESH_OBJECT

Explanation

This condition is detected when an object is refreshed.

Completion code

MQCC_WARNING

Programmer response

None. This reason code is only used to identify the corresponding event message “Refresh object” on page 225.

2371 (0943) (RC2371): MQRC_CHANNEL_SSL_ERROR

Explanation

This condition is detected when a connection cannot be established due to an TLS key-exchange or authentication failure.

Completion code

MQCC_WARNING

Programmer response

None. This reason code is only used to identify the corresponding event message “Channel SSL Error” on page 162.

z/OS 2373 (0945) (RC2373): MQRC_CF_STRUC_FAILED

Explanation

An MQI call or command was issued to access a shared queue, but the call failed because the coupling-facility structure used for the shared queue had failed.

This reason code occurs only on z/OS.

Completion code

MQCC_FAILED

Programmer response

Report the problem to the operator or administrator, who should use the MQSC command RECOVER CFSTRUCT to initiate recovery of the coupling-facility structure, unless automatic recovery has been enabled for the structure.

ALW 2374 (0946) (RC2374): MQRC_API_EXIT_ERROR

Explanation

An API exit function returned an invalid response code, or failed in some other way.

Completion code

MQCC_FAILED

Programmer response

Check the exit logic to ensure that the exit is returning valid values in the `ExitResponse` and `ExitResponse2` fields of the MQAXP structure. Consult the FFST record to see if it contains more detail about the problem.

ALW 2375 (0947) (RC2375): MQRC_API_EXIT_INIT_ERROR

Explanation

The queue manager encountered an error while attempting to initialize the execution environment for an API exit function.

Completion code

MQCC_FAILED

Programmer response

Consult the FFST record to obtain more detail about the problem.

ALW 2376 (0948) (RC2376): MQRC_API_EXIT_TERM_ERROR

Explanation

The queue manager encountered an error while attempting to terminate the execution environment for an API exit function.

Completion code

MQCC_FAILED

Programmer response

Consult the FFST record to obtain more detail about the problem.

ALW 2377 (0949) (RC2377): MQRC_EXIT_REASON_ERROR

Explanation

An MQXEP call was issued by an API exit function, but the value specified for the **ExitReason** parameter is either not valid, or not supported for the specified function identifier `Function`.

Completion code

MQCC_FAILED

Programmer response

Modify the exit function to specify a value for `ExitReason` that is valid for the specified value of `Function`.

ALW 2378 (094A) (RC2378): MQRC_RESERVED_VALUE_ERROR

Explanation

An MQXEP call was issued by an API exit function, but the value specified for the **Reserved** parameter is not valid. The value must be the null pointer.

Completion code

MQCC_FAILED

Programmer response

Modify the exit to specify the null pointer as the value of the **Reserved** parameter.

Multi 2379 (094B) (RC2379): MQRC_NO_DATA_AVAILABLE

Explanation

This reason should be returned by the MQZ_ENUMERATE_AUTHORITY_DATA installable service component when there is no more authority data to return to the invoker of the service component.

Completion code

MQCC_FAILED

Programmer response

None.

ALW 2380 (094C) (RC2380): MQRC_SCO_ERROR

Explanation

On an MQCONNX call, the MQSCO structure is not valid for one of the following reasons:

- The *StrucId* field is not MQSCO_STRUC_ID.
- The *Version* field specifies a value that is not valid or not supported.

Completion code

MQCC_FAILED

Programmer response

Correct the definition of the MQSCO structure.

ALW 2381 (094D) (RC2381): MQRC_KEY_REPOSITORY_ERROR

Explanation

On an MQCONN or MQCONNX call, the location of the key repository is either not specified, not valid, or results in an error when used to access the key repository.

The location of the key repository is specified by one of the following:

- The value of the **SSLKeyRepository** property in the SSL stanza of the client configuration file (MQCONN or MQCONNX call), or
- The value of the MQSSLKEYR environment variable (MQCONN or MQCONNX call), or
- The value of the KeyRepository field in the MQSCO structure (MQCONNX call only).

For the MQCONNX call, if both MQSSLKEYR and KeyRepository are specified, the latter is used.

This reason code can also occur if the passphrase to access the key repository is not supplied, incorrect, or cannot be decrypted. See Supplying the key repository password for an IBM MQ MQI clienton AIX, Linux, and Windows for more information.

Completion code

MQCC_FAILED

Programmer response

Ensure that a valid location is specified for the key repository.

Check the client error log for a message that indicates the cause of the error.

If the error log indicates that the key repository passphrase is incorrect, check that the correct key repository passphrase is specified, or that the key repository stash file exists and is readable by the user ID that the application is running under.

If the key repository passphrase is encrypted, and an initial key was specified to the **runmqicred** utility when the passphrase was encrypted, ensure that the initial key used to encrypt the passphrase is specified by the application.

ALW 2382 (094E) (RC2382): MQRC_CRYPTO_HARDWARE_ERROR

Explanation

On an MQCONN or MQCONNX call, the configuration string for the cryptographic hardware is not valid, or results in an error when used to configure the cryptographic hardware. The configuration string is specified by one of the following:

- The value of the MQSSLCRYP environment variable (MQCONN or MQCONNX call), or
- The value of the `CryptoHardware` field in the MQSCO structure (MQCONNX call only).

For the MQCONNX call, if both MQSSLCRYP and `CryptoHardware` are specified, the latter is used.

Completion code

MQCC_FAILED

Programmer response

Specify a valid configuration string for the cryptographic hardware.

ALW 2383 (094F) (RC2383): MQRC_AUTH_INFO_REC_COUNT_ERROR

Explanation

On an MQCONNX call, the `AuthInfoRecCount` field in the MQSCO structure specifies a value that is less than zero.

Completion code

MQCC_FAILED

Programmer response

Specify a value for `AuthInfoRecCount` that is zero or greater.

ALW 2384 (0950) (RC2384): MQRC_AUTH_INFO_REC_ERROR

Explanation

On an MQCONNX call, the MQSCO structure does not specify the address of the MQAIR records correctly. One of the following applies:

- `AuthInfoRecCount` is greater than zero, but `AuthInfoRecOffset` is zero and `AuthInfoRecPtr` is the null pointer.
- `AuthInfoRecOffset` is not zero and `AuthInfoRecPtr` is not the null pointer.
- `AuthInfoRecPtr` is not a valid pointer.
- `AuthInfoRecOffset` or `AuthInfoRecPtr` points to storage that is not accessible.

Completion code

MQCC_FAILED

Programmer response

Ensure that one of `AuthInfoRecOffset` or `AuthInfoRecPtr` is zero and the other nonzero. Ensure that the field used points to accessible storage.

ALW 2385 (0951) (RC2385): MQRC_AIR_ERROR

Explanation

On an MQCONNX call, an MQAIR record is not valid for one of the following reasons:

- The `StrucId` field is not MQAIR_STRUC_ID.
- The `Version` field specifies a value that is not valid or not supported.

Completion code

MQCC_FAILED

Programmer response

Correct the definition of the MQAIR record.

ALW 2386 (0952) (RC2386): MQRC_AUTH_INFO_TYPE_ERROR

Explanation

On an MQCONNX call, the *AuthInfoType* field in an MQAIR record specifies a value that is not valid.

Completion code

MQCC_FAILED

Programmer response

Specify MQAIT_CRL_LDAP for *AuthInfoType*.

ALW 2387 (0953) (RC2387): MQRC_AUTH_INFO_CONN_NAME_ERROR

Explanation

On an MQCONNX call, the AuthInfoConnName field in an MQAIR record specifies a value that is not valid.

Completion code

MQCC_FAILED

Programmer response

Specify a valid connection name.

ALW 2388 (0954) (RC2388): MQRC_LDAP_USER_NAME_ERROR

Explanation

On an MQCONNX call, an LDAP user name in an MQAIR record is not specified correctly. One of the following applies:

- LDAPUserNameLength is greater than zero, but LDAPUserNameOffset is zero and LDAPUserNamePtr is the null pointer.
- LDAPUserNameOffset is nonzero and *LDAPUserNamePtr* is not the null pointer.
- LDAPUserNamePtr is not a valid pointer.
- LDAPUserNameOffset or LDAPUserNamePtr points to storage that is not accessible.

Completion code

MQCC_FAILED

Programmer response

Ensure that one of LDAPUserNameOffset or LDAPUserNamePtr is zero and the other nonzero. Ensure that the field used points to accessible storage.

ALW 2389 (0955) (RC2389): MQRC_LDAP_USER_NAME_LENGTH_ERR

Explanation

On an MQCONNX call, the LDAPUserNameLength field in an MQAIR record specifies a value that is less than zero.

Completion code

MQCC_FAILED

Programmer response

Specify a value for LDAPUserNameLength that is zero or greater.

ALW 2390 (0956) (RC2390): MQRC_LDAP_PASSWORD_ERROR

Explanation

On an MQCONNX call, the LDAPPassword field in an MQAIR record specifies a value when no value is allowed.

Completion code

MQCC_FAILED

Programmer response

Specify a value that is blank or null.

ALW 2391 (0957) (RC2391): MQRC_SSL_ALREADY_INITIALIZED

Explanation

An MQCONN or MQCONNX call was issued when a connection is already open to the same queue manager. There is a conflict between the TLS options of the connections for one of three reasons:

- The TLS configuration options are different between the first and second connections.
- The existing connection was specified without TLS configuration options, but the second connection has TLS configuration options specified.
- The existing connection was specified with TLS configuration options, but the second connection does not have any TLS configuration options specified.

The connection to the queue manager completed successfully, but the TLS configuration options specified on the call were ignored; the existing TLS environment was used instead.

Completion code

MQCC_WARNING

Programmer response

If the application must be run with the TLS configuration options defined on the MQCONN or MQCONNX call, use the MQDISC call to sever the connection to the queue manager and then stop the application. Alternatively run the application later when the TLS environment has not been initialized.

ALW 2392 (0958) (RC2392): MQRC_SSL_CONFIG_ERROR

Explanation

On an MQCONNX call, the MQCNO structure does not specify the MQSCO structure correctly. One of the following applies:

- SSLConfigOffset is nonzero and SSLConfigPtr is not the null pointer.
- SSLConfigPtr is not a valid pointer.
- SSLConfigOffset or SSLConfigPtr points to storage that is not accessible.

Completion code

MQCC_FAILED

Programmer response

Ensure that one of `SSLConfigOffset` or `SSLConfigPtr` is zero and the other nonzero. Ensure that the field used points to accessible storage.

ALW 2393 (0959) (RC2393): MQRC_SSL_INITIALIZATION_ERROR

Explanation

An MQCONN or MQCONNX call was issued with TLS configuration options specified, but an error occurred during the initialization of the TLS environment.

Completion code

MQCC_FAILED

Programmer response

Check that the TLS installation is correct.

Look at the error logs for more information.

z/OS 2394 (095A) (RC2394): MQRC_Q_INDEX_TYPE_ERROR

Explanation

An MQGET call was issued specifying one or more of the following options:

- MQGMO_ALL_MSGS_AVAILABLE
- MQGMO_ALL_SEGMENTS_AVAILABLE
- MQGMO_COMPLETE_MSG
- MQGMO_LOGICAL_ORDER

but the call failed because the queue is not indexed by group identifier. These options require the queue to have an **IndexType** of `MQIT_GROUP_ID`.

This reason code occurs only on z/OS.

Completion code

MQCC_FAILED

Programmer response

Redefine the queue to have an **IndexType** of `MQIT_GROUP_ID`. Alternatively, modify the application to avoid using the options listed.

Multi 2395 (095B) (RC2395): MQRC_CFBS_ERROR

Explanation

An MQPUT or MQPUT1 call was issued, but the message data contains an MQCFBS structure that is not valid.

Completion code

MQCC_FAILED

Programmer response

Check that the fields in the structure are set correctly.

2396 (095C) (RC2396): MQRC_SSL_NOT_ALLOWED

Explanation

A connection to a queue manager was requested, specifying TLS encryption. However, the connection mode requested is one that does not support TLS (for example, bindings connect).

Completion code

MQCC_FAILED

Programmer response

Modify the application to request client connection mode, or to disable TLS encryption.

Note: Using a non null setting, including blanks, for the connection's cipher suite property can also cause this error.

2397 (095D) (RC2397): MQRC_JSSE_ERROR

Explanation

JSSE reported an error (for example, while connecting to a queue manager using TLS encryption). The MQException object containing this reason code references the Exception thrown by JSSE; this can be obtained by using the `MQException.getCause()` method. From JMS, the MQException is linked to the thrown JMSException.

This reason code occurs only with Java applications.

Completion code

MQCC_FAILED

Programmer response

Inspect the causal exception to determine the JSSE error.

2398 (095E) (RC2398): MQRC_SSL_PEER_NAME_MISMATCH

Explanation

The application attempted to connect to the queue manager using TLS encryption, but the distinguished name presented by the queue manager does not match the specified pattern.

Completion code

MQCC_FAILED

Programmer response

Check the certificates used to identify the queue manager. Also check the value of the **sslPeerName** property specified by the application.

2399 (095F) (RC2399): MQRC_SSL_PEER_NAME_ERROR

Explanation

The application specified a peer name of incorrect format.

Completion code

MQCC_FAILED

Programmer response

Check the value of the **sslPeerName** property specified by the application.

2400 (0960) (RC2400): MQRC_UNSUPPORTED_CIPHER_SUITE

Explanation

This reason code can occur in the following circumstances:

- A connection to a queue manager was requested, specifying TLS encryption. However, JSSE reported that it does not support the CipherSuite specified by the application.
- V 10.0.0 If an IBM MQ classes for Java or IBM MQ classes for JMS application, running in FIPS mode, attempts to connect to a queue manager that uses a non-FIPS compliant cipher.

This reason code occurs only with Java applications.

Completion code

MQCC_FAILED

Programmer response

Check the CipherSuite specified by the application. Note that the names of JSSE CipherSuites differ from their equivalent CipherSpecs used by the queue manager.

Also, check that JSSE is correctly installed.

2401 (0961) (RC2401): MQRC_SSL_CERTIFICATE_REVOKED

Explanation

A connection to a queue manager was requested, specifying TLS encryption. However, the certificate presented by the queue manager was found to be revoked by one of the specified CertStores.

This reason code occurs only with Java applications.

Completion code

MQCC_FAILED

Programmer response

Check the certificates used to identify the queue manager.

2402 (0962) (RC2402): MQRC_SSL_CERT_STORE_ERROR

Explanation

A connection to a queue manager was requested, specifying TLS encryption. However, none of the CertStore objects provided by the application could be searched for the certificate presented by the queue manager. The MQException object containing this reason code references the Exception encountered when searching the first CertStore; this can be obtained using the `MQException.getCause()` method. From JMS, the MQException is linked to the thrown JMSException.

This reason code occurs only with Java applications.

Completion code

MQCC_FAILED

Programmer response

Inspect the causal exception to determine the underlying error. Check the CertStore objects provided by your application. If the causal exception is a java.lang.NoSuchElementException, ensure that your application is not specifying an empty collection of CertStore objects.

2406 (0966) (RC2406): MQRC_CLIENT_EXIT_LOAD_ERROR

Explanation

The external user exit required for a client connection could not be loaded because the shared library specified for it cannot be found, or the entry point specified for it cannot be found.

This reason code occurs only with Java applications and .NET applications.

Completion code

MQCC_FAILED

Programmer response

Ensure that the correct library is specified and that the system PATH variable includes the relevant directory. Ensure also that the entry point is named properly and that the specified library exports it.

This error can occur if a Java application attempts to load a 32-bit native exit. To resolve the problem, convert the exit to Java code or run the exit in a 64-bit environment.

2407 (0967) (RC2407): MQRC_CLIENT_EXIT_ERROR

Explanation

A failure occurred while executing a non-Java user exit for a client connection.

This reason code occurs only with Java applications that use a non-Java exit.

Completion code

MQCC_FAILED

Programmer response

Check that the non-Java user exit can accept the parameters and message being passed to it and that it can handle error conditions, and that any information that the exit requires, such as user data, is correct and available.

2409 (0969) (RC2409): MQRC_SSL_KEY_RESET_ERROR

Explanation

On an MQCONN or MQCONNX call, the value of the TLS key reset count is not in the valid range of 0 through 999 999 999.

The value of the TLS key reset count is specified by either the value of the MQSSLRESET environment variable (MQCONN or MQCONNX call), or the value of the `KeyResetCount` field in the MQSCO structure (MQCONNX call only). For the MQCONNX call, if both MQSSLRESET and `KeyResetCount` are specified, the latter is used. MQCONN or MQCONNX

If you specify a TLS secret key reset count in the range 1 byte through 32Kb, TLS channels will use a secret key reset count of 32Kb. This is to avoid the overhead of excessive key resets which would occur for small TLS secret key reset values.

Completion code

MQCC_FAILED

Programmer response

Check that the fields in the structure and the MQSSLRESET environment variable are set correctly.

2411 (096B) (RC2411): MQRC_LOGGER_STATUS

Explanation

This condition is detected when a logger event occurs.

Completion code

MQCC_WARNING

Programmer response

None. This reason code is only used to identify the corresponding event message “Logger” on page 195.

2412 (096C) (RC2412): MQRC_COMMAND_MQSC

Explanation

This condition is detected when an MQSC command is executed.

Completion code

MQCC_WARNING

Programmer response

None. This reason code is only used to identify the corresponding event message “Command” on page 170.

2413 (096D) (RC2413): MQRC_COMMAND_PCF

Explanation

This condition is detected when a PCF command is executed.

Completion code

MQCC_WARNING

Programmer response

None. This reason code is only used to identify the corresponding event message “Command” on page 170.

2414 (096E) (RC2414): MQRC_CFIF_ERROR

Explanation

An MQPUT or MQPUT1 call was issued, but the message data contains an MQCFIF structure that is not valid.

Completion code

MQCC_FAILED

Programmer response

Check that the fields in the structure are set correctly.

2415 (096F) (RC2415): MQRC_CFSF_ERROR

Explanation

An MQPUT or MQPUT1 call was issued, but the message data contains an MQCFSF structure that is not valid.

Completion code

MQCC_FAILED

Programmer response

Check that the fields in the structure are set correctly.

2416 (0970) (RC2416): MQRC_CFGR_ERROR

Explanation

An MQPUT or MQPUT1 call was issued, but the message data contains an MQCFGR structure that is not valid.

Completion code

MQCC_FAILED

Programmer response

Check that the fields in the structure are set correctly.

2417 (0971) (RC2417): MQRC_MSG_NOT_ALLOWED_IN_GROUP

An explanation of the error, completion code, and programmer response.

Explanation

An MQPUT or MQPUT1 call was issued to put a message in a group but it is not valid to put such a message in a group. An example of an invalid message is a PCF message where the Type is MQCFT_TRACE_ROUTE.

You cannot use grouped or segmented messages with Publish/Subscribe.

Completion code

MQCC_FAILED

Programmer response

Remove the invalid message from the group.

2418 (0972) (RC2418): MQRC_FILTER_OPERATOR_ERROR

Explanation

The **Operator** parameter supplied is not valid.

If it is an input variable then the value is not one of the MQCFOP_* constant values. If it is an output variable then the parameter pointer is not valid, or it points to read-only storage. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredicatable results occur.)

Completion code

MQCC_FAILED

Programmer response

Correct the parameter.

2419 (0973) (RC2419): MQRC_NESTED_SELECTOR_ERROR

Explanation

An mqAddBag call was issued, but the bag to be nested contained a data item with an inconsistent selector. This reason only occurs if the bag into which the nested bag was to be added was created with the MQCBO_CHECK_SELECTORS option.

Completion code

MQCC_FAILED

Programmer response

Ensure that all data items within the bag to be nested have selectors that are consistent with the data type implied by the item.

2420 (0974) (RC2420): MQRC_EPH_ERROR

Explanation

An MQPUT or MQPUT1 call was issued, but the message data contains an MQEPH structure that is not valid. Possible errors include the following:

- The `StrucId` field is not MQEPH_STRUC_ID.
- The `Version` field is not MQEPH_VERSION_1.
- The `StrucLength` field specifies a value that is too small to include the structure plus the variable-length data at the end of the structure.
- The `CodedCharSetId` field is zero, or a negative value that is not valid.
- The `Flags` field contains an invalid combination of MQEPH_* values.
- The **`BufferLength`** parameter of the call has a value that is too small to accommodate the structure, so the structure extends beyond the end of the message.

Completion code

MQCC_FAILED

Programmer response

Check that the fields in the structure are set correctly. Ensure that the application sets the `CodedCharSetId` field to a valid value; note that MQCCSI_DEFAULT, MQCCSI_EMBEDDED, MQCCSI_Q_MGR, and MQCCSI_UNDEFINED are not valid in this field.

2421 (0975) (RC2421): MQRC_RFH_FORMAT_ERROR

Explanation

The message contains an MQRFH structure, but its format is incorrect.

Completion code

MQCC_FAILED

Programmer response

Correct the format in the RFH2 section of the IBM MQ request message, and retry.

Multi 2422 (0976) (RC2422): MQRC_CFBF_ERROR

Explanation

An MQPUT or MQPUT1 call was issued, but the message data contains an MQCFBF structure that is not valid.

Completion code

MQCC_FAILED

Programmer response

Check that the fields in the structure are set correctly.

2423 (0977) (RC2423): MQRC_CLIENT_CHANNEL_CONFLICT

Explanation

A client channel definition table (CCDT) was specified for determining the name of the channel, but the name has already been defined.

This reason code occurs only with Java applications.

Completion code

MQCC_FAILED

Programmer response

Change the channel name to blank and try again.

2424 (0978) (RC2424): MQRC_SD_ERROR

Explanation

On the MQSUB call, the Subscription Descriptor MQSD is not valid, for one of the following reasons:

- The StrucId field is not MQSD_SCTRUC_ID.
- The Version field specifies a value that is not valid or not supported.
- The parameter pointer is not valid (it is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results can occur).
- The queue manager cannot copy the changes structure to application storage, even though the call is successful. This can occur, for example, if the pointer points to read-only storage.

Completion code

MQCC_FAILED

Programmer response

Ensure that input fields in the MQSD structure are set correctly.

2425 (0979) (RC2425): MQRC_TOPIC_STRING_ERROR

Explanation

On the MQOPEN or MQPUT1 call in the Object Descriptor MQOD, or on the MQSUB call in the Subscription Descriptor MQSD the resultant full topic string is not valid.

One of the following applies:

- ObjectName contains the name of a TOPIC object with a TOPICSTR attribute that contains an empty topic string.
- The fully resolved topic string contains the escape character '%' and it is not followed by one of the characters, '*', '?' or '%', and the MQSO_WILDCARD_CHAR option has been used on an MQSUB call.
- On an MQOPEN, conversion cannot be performed using the CCSID specified in the MQOD structure.
- The topic string is greater than 255 characters when using IBM MQ Multicast messaging.

Completion code

MQCC_FAILED

Programmer response

Ensure that there are no invalid topic string characters in either ObjectString or ObjectName.

If using IBM MQ Multicast messaging, ensure that the topic string is less than 255 characters.

2426 (097A) (RC2426): MQRC_STS_ERROR

Explanation

On an MQSTAT call, the MQSTS structure is not valid, for one of the following reasons:

- The StrucId field is not MQSTS_STRUC_ID.
- The Version field specifies a value that is not valid or not supported.
- The parameter pointer is not valid. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)
- The queue manager cannot copy the changed structure to application storage, even though the call is successful. This can occur, for example, if the pointer points to read-only storage.

Completion code

MQCC_FAILED

Programmer response

Ensure that input fields in the MQSTS structure are set correctly.

2428 (097C) (RC2428): MQRC_NO_SUBSCRIPTION

Explanation

An MQSUB call using option MQSO_RESUME was made specifying a full subscription name that does not match any existing subscription.

Completion code

MQCC_FAILED

Programmer response

Ensure that the subscription exists and that the full subscription name is correctly specified in your application. The full subscription name is built from the ConnTag field specified at connection time in the MQCNO structure and the SubName field specified at MQSUB time in the MQSD structure.

2429 (097D) (RC2429): MQRC_SUBSCRIPTION_IN_USE

Explanation

An MQSUB call using option MQSO_RESUME was made specifying a full subscription name that is in use.

Completion code

MQCC_FAILED

Programmer response

Ensure that the subscription name is correctly specified in your application. The subscription name is specified in the SubName field in the MQSD structure.

2430 (097E) (RC2430): MQRC_STAT_TYPE_ERROR

Explanation

The STS parameter contains options that are not valid for the MQSTAT call. This reason also occurs if the parameter pointer is not valid. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)

Programmer response

Specify a valid MQSTS structure as a parameter on the call to MQSTAT.

2431 (097F) (RC2431): MQRC_SUB_USER_DATA_ERROR

Explanation

On the MQSUB call in the Subscription Descriptor MQSD the SubUserData field is not valid. One of the following applies:

- SubUserData.VSLength is greater than zero, but SubUserData.VSOffset is zero and SubUserData.VSPtr is the null pointer.
- SubUserData.VSOffset is nonzero and SubUserData.VSPtr is not the null pointer (that is, it appears both fields are being used where only one is allowed).
- SubUserData.VSPtr is not a valid pointer.
- SubUserData.VSOffset or SubUserData.VSPtr points to storage that is not accessible.
- SubUserData.VSLength exceeds the maximum length allowed for this field.

Completion code

MQCC_FAILED

Programmer response

Ensure that one of `SubUserData.VSOffset` or `SubUserData.VSPtr` is zero and the other nonzero. Ensure that the field used points to accessible storage. Specify a length that does not exceed the maximum length allowed for this field.

2432 (0980) (RC2432): MQRC_SUB_ALREADY_EXISTS

Explanation

An MQSUB call was issued to create a subscription, using the MQSO_CREATE option, but a subscription using the same SubName and ObjectString already exists.

Completion code

MQCC_FAILED

Programmer response

Ensure that the SubName and ObjectString input fields in the MQSD structure are set correctly, or use the MQSO_RESUME option to get a handle for the subscription that already exists.

2434 (0982) (RC2434): MQRC_IDENTITY_MISMATCH

Explanation

An MQSUB call using either MQSO_RESUME or MQSO_ALTER was made against a subscription that has the MQSO_FIXED_USERID option set, by a userid other than the one recorded as owning the subscription.

Completion code

MQCC_FAILED

Programmer response

Correct the full subscription name to one that is unique, or update the existing subscription to allow different userids to use it by using the MQSO_ANY_USERID option from an application running under the owning userid.

2435 (0983) (RC2435): MQRC_ALTER_SUB_ERROR

Explanation

An MQSUB call using option MQSO_ALTER was made changing a subscription that was created with the MQSO_IMMUTABLE option.

Completion code

MQCC_FAILED

Programmer response

Remove the subscription using MQCLOSE and re-create it with MQSUB with the attributes set correctly.

2436 (0984) (RC2436): MQRC_DURABILITY_NOT_ALLOWED

Explanation

An MQSUB call using the MQSO_DURABLE option failed. This can be for one of the following reasons:

- The topic subscribed to is defined as **DURSUB(NO)**.
- The queue named SYSTEM.DURABLE.SUBSCRIBER.QUEUE is not available.
- The topic subscribed to is defined as both **MCAST(ONLY)** and **DURSUB(YES)** (or **DURSUB(ASPARENT)** and the parent is **DURSUB(YES)**).

Completion code

MQCC_FAILED

Programmer response

Durable subscriptions are stored on the SYSTEM.DURABLE.SUBSCRIBER.QUEUE. Ensure that this queue is available for use. Possible reasons for failure include the queue being full, the queue being put inhibited, the queue not existing.

z/OS On z/OS, another possible reason for failure is that the page set the queue is defined to use does not exist.

If the topic subscribed to is defined as **DURSUB(NO)**, either alter the administrative topic node to use **DURSUB(YES)** or use the MQSO_NON_DURABLE option instead.

If the topic subscribed to is defined as **MCAST(ONLY)** when using IBM MQ Multicast messaging, alter the topic to use **DURSUB(NO)**.

2437 (0985) (RC2437): MQRC_NO_RETAINED_MSG

Explanation

An MQSUBRQ call was made to a topic to request that any retained publications for this topic are sent to the subscriber. However, there are no retained publications currently stored for this topic.

Completion code

MQCC_FAILED

Programmer response

Ensure that publishers to the topic are marking their publication to be retained and that publications are being made to this topic.

2438 (0986) (RC2438): MQRC_SRO_ERROR

Explanation

On the MQSUBRQ call, the Subscription Request Options MQSRO is not valid, for one of the following reasons:

- The `StrucId` field is not MQSRO_STRUC_ID.
- The `Version` field specifies a value that is not valid or not supported.

- The parameter pointer is not valid. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)
- The queue manager cannot copy the changed structure to application storage, even though the call is successful. This can occur, for example, if the pointer points to read-only storage.

Completion code

MQCC_FAILED

Programmer response

Ensure that input fields in the MQSRO structure are set correctly.

2440 (0988) (RC2440): MQRC_SUB_NAME_ERROR

Explanation

On the MQSUB call in the Subscription Descriptor MQSD the SubName field is not valid or has been omitted. This is required if the MQSD option MQSO_DURABLE is specified, but may also be used if MQSO_DURABLE is not specified.

One of the following applies:

- SubName.VSLength is greater than zero, but SubName.VSOffset is zero and SubName.VSPtr is the null pointer.
- SubName.VSOffset is nonzero and SubName.VSPtr is not the null pointer (that is, it appears both fields are being used where only one is allowed).
- SubName.VSPtr is not a valid pointer.
- SubName.VSOffset or SubName.VSPtr points to storage that is not accessible.
- SubName.VSLength is zero but this field is required.
- SubName.VSLength exceeds the maximum length allowed for this field.

Completion code

MQCC_FAILED

Programmer response

Ensure that SubName is specified and SubName.VSLength is nonzero. Ensure that one of SubName.VSOffset or SubName.VSPtr is zero and the other nonzero. Ensure that the field used points to accessible storage. Specify a length that does not exceed the maximum length allowed for this field.

This code can be returned if the sd.Options flags MQSO_CREATE and MQSO_RESUME are set together and sd.SubName is not initialized. You must also initialize the MQCHARV structure for sd.SubName, even if there is no subscription to resume; see Example 2: Managed MQ subscriber for more details.

2441 (0989) (RC2441): MQRC_OBJECT_STRING_ERROR

Explanation

On the MQOPEN or MQPUT1 call in the Object Descriptor MQOD, or on the MQSUB call in the Subscription Descriptor MQSD the ObjectString field is not valid.

One of the following applies:

- ObjectString.VSLength is greater than zero, but ObjectString.VSOffset is zero and ObjectString.VSPtr is the null pointer.

- `ObjectString.VSOffset` is nonzero and `ObjectString.VSPtr` is not the null pointer (that is, it appears both fields are being used where only one is allowed).
- `ObjectString.VSPtr` is not a valid pointer.
- `ObjectString.VSOffset` or `ObjectString.VSPtr` points to storage that is not accessible.
- `ObjectString.VSLength` exceeds the maximum length allowed for this field.

Completion code

MQCC_FAILED

Programmer response

Ensure that one of `ObjectString.VSOffset` or `ObjectString.VSPtr` is zero and the other nonzero. Ensure that the field used points to accessible storage. Specify a length that does not exceed the maximum length allowed for this field.

2442 (098A) (RC2442): MQRC_PROPERTY_NAME_ERROR

Explanation

An attempt was made to set a property with an invalid name. Using any of the following settings results in this error:

- The name contains an invalid character.
- The name begins "JMS" or "usr.JMS" and the JMS property is not recognized.
- The name begins "mq" in any mixture of lowercase or uppercase and is not "mq_usr" and contains more than one "." character (U+002E). Multiple "." characters are not allowed in properties with those prefixes.
- Characters beyond the Unicode Basic Multilingual Plane (those above U+FFFF), represented in UTF-16 by surrogate code points (X'D800' through X'DFFF'), or four bytes in UTF-8, are not supported in message property names.
- The name is "NULL", "TRUE", "FALSE", "NOT", "AND", "OR", "BETWEEN", "LIKE", "IN", "IS" and "ESCAPE" or is one of these keywords prefixed by "usr.".
- The name begins with "Body" or "Root" (except for names beginning "Root.MQMD.").
- A "." character must not be followed immediately by another "." character.
- The "." character cannot be the last character in a property name.

Completion code

MQCC_FAILED

Programmer response

Valid property names are described in the IBM MQ documentation. Ensure that all properties in the message have valid names before reissuing the call.

2443 (098B) (RC2443): MQRC_SEGMENTATION_NOT_ALLOWED

Explanation

An MQPUT or MQPUT1 call was issued to put a segmented message or a message that may be broken up into smaller segments (MQMF_SEGMENTATION_ALLOWED). The message was found to contain one or more MQ-defined properties in the message data; MQ-defined properties are not valid in the message data of a segmented message.

IBM MQ Multicast cannot use segmented messages.

Completion code

MQCC_FAILED

Programmer response

Remove the invalid properties from the message data or prevent the message from being segmented.

2444 (098C) (RC2444): MQRC_CBD_ERROR

Explanation

An MQCB call the MQCBD structure is not valid for one of the following reasons:

- The `StrucId` field is not MQCBD_STRUC_ID
- The `Version` field is specifies a value that is not valid or is not supported
- The parameter pointer is not valid. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)

Completion code

MQCC_FAILED

Programmer response

Ensure that input fields in the MQCBD structure are set correctly.

2445 (098D) (RC2445): MQRC_CTLO_ERROR

Explanation

On a MQCTL call the MQCTLO structure is not valid for one of the following reasons:

- The `StrucId` field is not MQCTLO_STRUC_ID
- The `Version` field is specifies a value that is not valid or is not supported
- The parameter pointer is not valid. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)

Completion code

MQCC_FAILED

Programmer response

Ensure that input fields in the MQCTLO structure are set correctly.

2446 (098E) (RC2446): MQRC_NO_CALLBACKS_ACTIVE

Explanation

An MQCTL call was made with an Operation of MQOP_START_WAIT and has returned because there are no currently defined callbacks which are not suspended.

Completion code

MQCC_FAILED

Programmer response

Ensure that there is at least one registered, resumed consumer function.

2448 (0990) (RC2448): MQRC_CALLBACK_NOT_REGISTERED

Explanation

An attempt to issue an MQCB call has been made against an object handle which does not currently have a registered callback.

Completion code

MQCC_FAILED

Programmer response

Ensure that a callback has been registered against the object handle.

2449 (0991) (RC2449): MQRC_OPERATION_NOT_ALLOWED

Explanation

An MQCTL call was made with an Operation that is not allowed, because of the current state of asynchronous consumption on the hConn.

If Operation was MQOP_RESUME, the operation is not allowed because the state of asynchronous consumption on the hConn is STOPPED. Re-issue MQCTL with the MQOP_START Operation.

If Operation was MQOP_SUSPEND, the operation is not allowed because the state of asynchronous consumption on the hConn is STOPPED. If you need to get your hConn into a SUSPENDED state, issue MQCTL with the MQOP_START Operation followed by MQCTL with MQOP_SUSPEND.

If Operation was MQOP_START, the operation is not allowed because the state of asynchronous consumption on the hConn is SUSPENDED. Re-issue MQCTL with the MQOP_RESUME Operation.

If Operation was MQOP_START_WAIT, the operation is not allowed because either

- The state of asynchronous consumption on the hConn is SUSPENDED. Re-issue MQCTL with the MQOP_RESUME Operation.
- The state of asynchronous consumption on the hConn is already STARTED. Do not mix the use of MQOP_START and MQOP_START_WAIT within one application.

Completion code

MQCC_FAILED

Programmer response

Re-issue the MQCTL call with the correct Operation.

2457 (0999) (RC2457): MQRC_OPTIONS_CHANGED

Explanation

An MQGET call on a queue handle opened using MQOO_READ_AHEAD (or resolved to that value through the queue's default value) has altered an option that is required to be consistent between MQGET calls.

Completion code

MQCC_FAILED

Programmer response

Keep all required MQGET options the same between invocations of MQGET, or use MQOO_NO_READ_AHEAD when opening the queue. For more information, see MQGET options and read ahead.

2458 (099A) (RC2458): MQRC_READ_AHEAD_MSGS

Explanation

On an MQCLOSE call, the option MQCO_QUIESCE was used and there are still messages stored in client read ahead buffer that were sent to the client ahead of an application requesting them and have not yet been consumed by the application.

Completion code

MQCC_WARNING

Programmer response

Continue to consume messages using the queue handle until there are no more available and then issue the MQCLOSE again, or choose to discard these messages by issuing the MQCLOSE call with the MQCO_IMMEDIATE option instead.

2459 (099B) (RC2459): MQRC_SELECTOR_SYNTAX_ERROR

Explanation

An MQOPEN, MQPUT1 or MQSUB call was issued but a selection string was specified which contained a syntax error.

Completion code

MQCC_FAILED

Programmer response

See Message selector syntax and ensure that you have correctly followed the rules for specifying selection strings. Correct any syntax errors and resubmit the MQ API call for which the error occurred.

2460 (099C) (RC2460): MQRC_HMSG_ERROR

Explanation

On an MQCRTMH, MQDLTMH, MQSETMP, MQINQMP or MQDLT call, a message handle supplied is not valid, for one of the following reasons:

- The parameter pointer is not valid, or (for the MQCRTMH call) points to read-only storage. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)
- The value specified was not returned by a preceding MQCRTMH call.
- The value specified has been made invalid by a preceding MQDLTMH call.

Completion code

MQCC_FAILED

Programmer response

Ensure that a successful MQCRTMH call is performed for the connection, and that an MQDLTMH call has not already been performed for it. Ensure that the handle is being used within its valid scope, for more information, see MQCRTMH - Create message handle.

2461 (099D) (RC2461): MQRC_CMHO_ERROR

Explanation

On an MQCRTMH call, the create message handle options structure MQCMHO is not valid, for one of the following reasons:

- The `StrucId` field is not MQCMHO_STRUC_ID.
- The `Version` field specifies a value that is not valid or not supported.
- The parameter pointer is not valid. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)

Completion code

MQCC_FAILED

Programmer response

Ensure that input fields in the MQCMHO structure are set correctly.

2462 (099E) (RC2462): MQRC_DMHO_ERROR

Explanation

On an MQDLTMH call, the delete message handle options structure MQDMHO is not valid, for one of the following reasons:

- The `StrucId` field is not MQCMHO_STRUC_ID.
- The `Version` field specifies a value that is not valid or not supported.
- The parameter pointer is not valid. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)

Completion code

MQCC_FAILED

Programmer response

Ensure that input fields in the MQDMHO structure are set correctly.

2463 (099F) (RC2463): MQRC_SMPO_ERROR

Explanation

On an MQSETMP call, the set message property options structure MQSMPO is not valid, for one of the following reasons:

- The `StrucId` field is not MQSMPO_STRUC_ID.
- The `Version` field specifies a value that is not valid or not supported.
- The parameter pointer is not valid. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)

Completion code

MQCC_FAILED

Programmer response

Ensure that input fields in the MQSMPO structure are set correctly.

2464 (09A0) (RC2464): MQRC_IMPO_ERROR

Explanation

On an MQINQMP call, the inquire message property options structure MQIMPO is not valid, for one of the following reasons:

- The `StrucId` field is not MQIMPO_STRUC_ID.
- The `Version` field specifies a value that is not valid or not supported.
- The parameter pointer is not valid. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)
- The queue manager cannot copy the changed structure to application storage, even though the call is successful. This can occur, for example, if the pointer points to read-only storage.

Completion code

MQCC_FAILED

Programmer response

Ensure that input fields in the MQIMPO structure are set correctly.

2465 (09A1) (RC2465): MQRC_PROPERTY_NAME_TOO_BIG

Explanation

On an MQINQMP call, IBM MQ attempted to copy the name of the inquired property into the location indicated by the ReturnedName field of the **InqPropOpts** parameter but the buffer was too small to contain the full property name. The call failed but the VSLength field of the ReturnedName of the **InqPropOpts** parameter indicates how large the ReturnedName buffer needs to be.

Completion code

MQCC_FAILED

Programmer response

The full property name can be retrieved by calling MQINQMP again with a larger buffer for the returned name, also specifying the MQIMPO_INQ_PROP_UNDER_CURSOR option. This will inquire on the same property.

2466 (09A2) (RC2466): MQRC_PROP_VALUE_NOT_CONVERTED

Explanation

An MQINQMP call was issued with the MQIMPO_CONVERT_VALUE option specified in the **InqPropOpts** parameter, but an error occurred during conversion of the value of the property. The property value is returned unconverted, the values of the ReturnedCCSID and ReturnedEncoding fields in the **InqPropOpts** parameter are set to those of the value returned.

Completion code

MQCC_FAILED

Programmer response

Check that the property value is correctly described by the **ValueCCSID** and **ValueEncoding** parameters that were specified when the property was set. Also check that these values, and the RequestedCCSID and RequestedEncoding specified in the **InqPropOpts** parameter of the MQINQMP call, are supported for IBM MQ conversion. If the required conversion is not supported, conversion must be carried out by the application.

2467 (09A3) (RC2467): MQRC_PROP_TYPE_NOT_SUPPORTED

Explanation

An MQINQMP call was issued and the property inquired has an unsupported data type. A string representation of the value is returned and the TypeString field of the **InqPropOpts** parameter can be used to determine the data type of the property.

Completion code

MQCC_WARNING

Programmer response

Check whether the property value was intended to have a data type indicated by the `TypeString` field. If so, the application must decide how to interpret the value. If not, modify the application that set the property to give it a supported data type.

2469 (09A5) (RC2469): MQRC_PROPERTY_VALUE_TOO_BIG

Explanation

On an MQINQMP call, the property value was too large to fit into the supplied buffer. The `DataLength` field is set to the length of the property value before truncation and the **`Value`** parameter contains as much of the value as fits.

On an MQMHBUF call, the `BufferLength` was less than the size of the properties to be put in the buffer. In this case the call fails. The `DataLength` field is set to the length of the properties before truncation.

Completion code

MQCC_WARNING

MQCC_FAILED

Programmer response

Supply a buffer that is at least as large as DataLength if all of the property value data is required and call MQINQMP again with the MQIMPO_INQ_PROP_UNDER_CURSOR option specified.

2470 (09A6) (RC2470): MQRC_PROP_CONV_NOT_SUPPORTED

Explanation

On an MQINQMP call, the MQIMPO_CONVERT_TYPE option was specified to request that the property value be converted to the supplied data type before the call returned. Conversion between the actual and requested property data types is not supported. The **Type** parameter indicates the data type of the property value.

Completion code

MQCC_FAILED

Programmer response

Either call MQINQMP again without MQIMPO_CONVERT_TYPE specified, or request a data type for which conversion is supported.

2471 (09A7) (RC2471): MQRC_PROPERTY_NOT_AVAILABLE

Explanation

On an MQINQMP call, no property could be found that matched the specified name. When iterating through multiple properties, possibly using a name containing a wildcard character, this indicates that all properties matching the name have now been returned.

Completion code

MQCC_FAILED

Programmer response

Ensure that the correct property name was specified. If the name contains a wildcard character specify option MQIMPO_INQ_FIRST to begin iterating over the properties again.

2472 (09A8) (RC2472): MQRC_PROP_NUMBER_FORMAT_ERROR

Explanation

On an MQINQMP call, conversion of the property value was requested. The format of the property is invalid for conversion to the requested data type.

Completion code

MQCC_FAILED

Programmer response

Ensure that the correct property name and data type were specified. Ensure that the application setting the property gave it the correct format. See the documentation for the MQINQMP call for details on the formats required for data conversion of property values.

2473 (09A9) (RC2473): MQRC_PROPERTY_TYPE_ERROR

Explanation

On an MQSETMP call, the Type parameter does not specify a valid MQTYPE_* value. For properties beginning "Root.MQMD." or "JMS" the specified Type must correspond to the data type of the matching MQMD or JMS header field:

- For MQCHARn or Java String fields use MQTYPE_STRING.
- For MQLONG or Java int fields use MQTYPE_INT32.
- For MQBYTEn fields use MQTYPE_BYTE_STRING.
- For Java long fields use MQTYPE_INT64.

On an MQINQMP call, the **Type** parameter is not valid. Either the parameter pointer is not valid, the value is invalid, or it points to read-only storage. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)

Completion code

MQCC_FAILED

Programmer response

Correct the parameter.

2478 (09AE) (RC2478): MQRC_PROPERTIES_TOO_BIG

Explanation

An MQPUT or MQPUT1 call was issued to put a message on a queue, but the properties of the message were too large. The length of the properties cannot exceed the value of the **MaxPropertiesLength** queue manager attribute. This return code will also be issued if a message with headers greater than 511 KB is put to a shared queue.

Completion code

MQCC_FAILED

Programmer response

Consider one of the following actions:

- Reduce the number or the size of the properties associated with the message. This could include moving some of the properties into the application data.
- Increase the value of the MaxPropertiesLength queue manager attribute.

2479 (09AF) (RC2479): MQRC_PUT_NOT_RETAINED

Explanation

An MQPUT or MQPUT1 call was issued to publish a message on a topic, using the MQPMO_RETAIN option, but the publication was unable to be retained. The publication is not published to any matching subscribers.

Completion code

MQCC_FAILED

Programmer response

Retained publications are stored on the SYSTEM.RETAINED.PUB.QUEUE. Ensure that this queue is available for use by the application. Possible reasons for failure include the queue being full, the queue being put inhibited, or the queue not existing.

2480 (09B0) (RC2480): MQRC_ALIAS_TARGTYPE_CHANGED

Explanation

An MQPUT or MQPUT1 call was issed to publish a message on a topic. One of the subscriptions matching this topic was made with a destination queue that was an alias queue which originally referenced a queue, but now references a topic object, which is not allowed. In this situation the reason code MQRC_ALIAS_TARGTYPE_CHANGED is returned in the Feedback field in the MQMD of a report message, or in the Reason field in the MQDLH structure of a message on the dead-letter queue.

Completion code

MQCC_FAILED

Programmer response

Find the subscriber that is using an alias queue which references a topic object and change it to reference a queue again, or change the subscription to reference a different queue.

2481 (09B1) (RC2481): MQRC_DMPO_ERROR

Explanation

On an MQDLTMP call, the delete message property options structure MQDMPO is not valid, for one of the following reasons:

- The StrucId field is not MQDMPO_STRUC_ID.

- The Version field specifies a value that is not valid or not supported.
- The parameter pointer is not valid. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)

Completion code

MQCC_FAILED

Programmer response

Ensure that input fields in the MQDMPO structure are set correctly.

2482 (09B2) (RC2482): MQRC_PD_ERROR

Explanation

On an MQSETMP or MQINQMP call, the property descriptor structure MQPD is not valid, for one of the following reasons:

- The StrucId field is not MQPD_STRUC_ID.
- The Version field specifies a value that is not valid or not supported.
- The parameter pointer is not valid. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)
- The Context field contains an unrecognized value.

Completion code

MQCC_FAILED

Programmer response

Ensure that input fields in the MQPD structure are set correctly.

2483 (09B3) (RC2483): MQRC_CALLBACK_TYPE_ERROR

Explanation

An MQCB call was made with an Operation of MQOP_REGISTER with an incorrect value for CallbackType.

Completion code

MQCC_FAILED

Programmer response

Ensure that the CallbackType field of the MQCBDO is specified correctly.

2484 (09B4) (RC2484): MQRC_CBD_OPTIONS_ERROR

Explanation

An MQCB call was made with an Operation of MQOP_REGISTER with an incorrect value for the Options field of the MQCBD.

Completion code

MQCC_FAILED

Programmer response

Ensure that the `Options` are specified correctly.

2485 (09B5) (RC2485): MQRC_MAX_MSG_LENGTH_ERROR

Explanation

An MQCB call was made with an Operation of MQOP_REGISTER with an incorrect value for the `MaxMsgLength` field of the MQCBD.

Completion code

MQCC_FAILED

Programmer response

Ensure that the `MaxMsgLength` is specified correctly.

2486 (09B6) (RC2486): MQRC_CALLBACK_ROUTINE_ERROR

Explanation

An MQCB call was made with an Operation of MQOP_REGISTER failed for one of the following reasons:

- Both `CallbackName` and `CallbackFunction` are specified. Only one must be specified on the call.
- The call was made from an environment not supporting function pointers.
- A programming language that does not support function pointer references.

Completion code

MQCC_FAILED

Programmer response

Ensure that the `CallbackName` value is specified correctly.

z/OS 2487 (09B7) (RC2487): MQRC_CALLBACK_LINK_ERROR

Explanation

On an MQCTL call, the callback handling module (CSQBMCSM or CSQBMCSX for batch and DFHMQMCM for CICS) could not be loaded, so the adapter could not link to it.

This reason code occurs only on z/OS.

Completion code

MQCC_FAILED

Programmer response

Ensure that the correct library concatenation has been specified in the application program execution JCL, and in the queue manager startup JCL. Any uncommitted changes in a unit of work should be backed out. A unit of work that is coordinated by the queue manager is backed out automatically.

2488 (09B8) (RC2488): MQRC_OPERATION_ERROR

Explanation

An MQCTL or MQCB call was made with an invalid parameter.

There is a conflict with the value specified for **Operation** parameter.

This error can be caused by an invalid value in the **Operation** parameter, no registered consumers when using MQOP_START or MQOP_START_WAIT parameter, and trying to use non-threaded libraries with asynchronous API calls.

Completion code

MQCC_FAILED

Programmer response

Investigate the application program and verify the **Operation** parameter options are correct. Ensure you have link edited the application with the correct version of the threading libraries for asynchronous functions.

2489 (09B9) (RC2489): MQRC_BMHO_ERROR

Explanation

On an MQBUFMH call, the buffer to message handle options structure MQBMHO is not valid, for one of the following reasons:

- The `StrucId` field is not MQBMHO_STRUC_ID.
- The `Version` field specifies a value that is not valid or not supported.
- The parameter pointer is not valid. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)

Completion code

MQCC_FAILED

Programmer response

Ensure that input fields in the MQBMHO structure are set correctly.

2490 (09BA) (RC2490): MQRC_UNSUPPORTED_PROPERTY

Explanation

A message was found to contain a property that the queue manager does not support. The operation that failed required all the properties to be supported by the queue manager. This can occur on the MQPUT/MQPUT1 call or when a message is about to be sent down a channel to a queue manager than does not support message properties.

Completion code

MQCC_FAILED

Programmer response

Determine which property of the message is not supported by the queue manager and decide whether to remove the property from the message or connect to a queue manager which does support the property.

V 9.4.1 V 9.4.0.5 2491 0x000009bb MQRC_MSG_LENGTH_ERROR

Explanation

A message was put to a remote queue but the channel was unable to process the full message from the transmission queue because the message has properties that cause its total size to exceed the maximum message length that is allowed by the queue manager. The message has been moved to the dead-letter queue.

Completion code

MQCC_FAILED

Programmer response

Where possible, use your dead-letter queue handler to process any such messages from the dead-letter queue.

If you have not configured a dead-letter queue handler, you can write an application to process the message. The application must use MQGET with option MQGMO_PROPERTIES_IN_HANDLE to get the message with properties in a message handle and the message payload in the message buffer.

2492 (09BC) (RC2492): MQRC_PROP_NAME_NOT_CONVERTED

Explanation

An MQINQMP call was issued with the MQIMPO_CONVERT_VALUE option specified in the **InqPropOpts** parameter, but an error occurred during conversion of the returned name of the property. The returned name is unconverted

Completion code

MQCC_WARNING

Programmer response

Check that the character set of the returned name was correctly described when the property was set. Also check that these values, and the `RequestedCCSID` and `RequestedEncoding` specified in the **InqPropOpts** parameter of the MQINQMP call, are supported for IBM MQ conversion. If the required conversion is not supported, conversion must be carried out by the application.

2494 (09BE) (RC2494): MQRC_GET_ENABLED

Explanation

This reason code is returned to an asynchronous consumer at the time a queue that was previously inhibited for get has been re-enabled for get.

Completion code

MQCC_WARNING

Programmer response

None. This reason code is used to inform the application of the change in state of the queue.

2495 (09BF) (RC2495): MQRC_MODULE_NOT_FOUND

Explanation

A native shared library could not be loaded.

Completion code

MQCC_FAILED

Programmer response

This problem could be caused by either of the two following reasons:

- A MQCB call was made with an Operation of MQOP_REGISTER specifying a `CallbackName` which could not be found. Ensure that the `CallbackName` value is specified correctly.
- The Java IBM MQ code could not load a Java native shared library. Check the associated Exception stack and FFST. Ensure that the JNI shared library is specified correctly. Check also that you have specified `-Djava.library.path=/opt/mqm/java/lib`, or equivalent, when invoking the Java program.

Related reference

The Java Native Interface (JNI) libraries required by IBM MQ classes for JMS applications

2496 (09C0) (RC2496): MQRC_MODULE_INVALID

Explanation

An MQCB call was made with an Operation of MQOP_REGISTER, specifying a `CallbackName` which is not a valid load module.

Completion code

MQCC_FAILED

Programmer response

Ensure that the `CallbackName` value is specified correctly.

2497 (09C1) (RC2497): MQRC_MODULE_ENTRY_NOT_FOUND

Explanation

An MQCB call was made with an Operation of MQOP_REGISTER and the `CallbackName` identifies a function name which can't be found in the specified library.

Completion code

MQCC_FAILED

Programmer response

Ensure that the `CallbackName` value is specified correctly.

2498 (09C2) (RC2498): MQRC_MIXED_CONTENT_NOT_ALLOWED

Explanation

An attempt was made to set a property with mixed content. For example, if an application set the property "x.y" and then attempted to set the property "x.y.z" it is unclear whether in the property name hierarchy "y" contains a value or another logical grouping. Such a hierarchy would be "mixed content" and this is not supported. Setting a property which would cause mixed content is not allowed. A hierarchy within a property name is created using the "." character (U+002E).

Completion code

MQCC_FAILED

Programmer response

Valid property names are described in Property names. Change the property name hierarchy so that it no longer contains mixed content before re-issuing the call.

2499 (09C3) (RC2499): MQRC_MSG_HANDLE_IN_USE

Explanation

A message property call was called (MQCRTMH, MQDLTMH, MQSETMP, MQINQMP, MQDLTMP or MQMHBUF) specifying a message handle that is already in use on another API call. A message handle may only be used on one call at a time.

Concurrent use of a message handle can arise, for example, when an application uses multiple threads.

Completion code

MQCC_FAILED

Programmer response

Ensure that the message handle cannot be used while another call is in progress.

2500 (09C4) (RC2500): MQRC_HCONN_ASYNC_ACTIVE

Explanation

An attempt to issue an MQI call has been made while the connection is started.

Completion code

MQCC_FAILED

Programmer response

Stop or suspend the connection using the MQCTL call and retry the operation.

2501 (09C5) (RC2501): MQRC_MHBO_ERROR

Explanation

On an MQMHBUF call, the message handle to buffer options structure MQMHBO is not valid, for one of the following reasons:

- The `StrucId` field is not MQMHBO_STRUC_ID.
- The `Version` field specifies a value that is not valid or not supported.
- The parameter pointer is not valid. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)

Completion code

MQCC_FAILED

Programmer response

Ensure that input fields in the MQMHBO structure are set correctly.

2502 (09C6) (RC2502): MQRC_PUBLICATION_FAILURE

Explanation

An MQPUT or MQPUT1 call was issued to publish a message on a topic. Delivery of the publication to one of the subscribers failed and due to the combination of the syncpoint option used and either:

- The PMSGDLV attribute on the administrative TOPIC object if it was a persistent message.
- The NPMSGDLV attribute on the administrative TOPIC object if it was a non-persistent message.

The publication has not been delivered to any of the subscribers.

Completion code

MQCC_FAILED

Programmer response

Find the subscriber or subscribers who are having problems with their subscription queue and resolve the problem, or change the setting of the PMSGDLV or NPMSGDLV attributes on the TOPIC so that problems with one subscriber do not have an effect on other subscribers. Retry the MQPUT.

2503 (09C7) (RC2503): MQRC_SUB_INHIBITED

Explanation

MQSUB calls are currently inhibited for the topic subscribed to.

Completion code

MQCC_FAILED

Programmer response

If the system design allows subscription requests to be inhibited for short periods, retry the operation later.

2504 (09C8) (RC2504): MQRC_SELECTOR_ALWAYS_FALSE

Explanation

An MQOPEN, MQPUT1 or MQSUB call was issued but a selection string was specified which will never select a message

Completion code

MQCC_FAILED

Programmer response

Verify that the logic of the selection string which was passed in on the API is as expected. Make any necessary corrections to the logic of the string and resubmit the MQ API call for which the message occurred.

2507 (09CB) (RC2507): MQRC_XEPO_ERROR

Explanation

On an MQXEP call, the exit options structure MQXEPO is not valid, for one of the following reasons:

- The `StrucId` field is not MQXEPO_STRUC_ID.
- The `Version` field specifies a value that is not valid or not supported.
- The parameter pointer is not valid. (It is not always possible to detect parameter pointers that are not valid; if not detected, unpredictable results occur.)

Completion code

MQCC_FAILED

Programmer response

Ensure that input fields in the MQXEPO structure are set correctly.

2509 (09CD) (RC2509): MQRC_DURABILITY_NOT_ALTERABLE

Explanation

An MQSUB call using option MQSO_ALTER was made changing the durability of the subscription. The durability of a subscription cannot be changed.

Completion code

MQCC_FAILED

Programmer response

Remove the subscription using MQCLOSE and re-create it with MQSUB with the attributes set correctly, or change the durability option used on the MQSUB call so that it matches the existing subscription.

2510 (09CE) (RC2510): MQRC_TOPIC_NOT_ALTERABLE

Explanation

An MQSUB call using option MQSO_ALTER was made changing the one or more of the fields in the MQSD that provide the topic being subscribed to. These fields are the `ObjectName`, `ObjectString`, or wildcard options. The topic subscribed to cannot be changed.

Completion code

MQCC_FAILED

Programmer response

Remove the subscription using MQCLOSE and re-create it with MQSUB with the attributes set correctly, or change the attributes and options used on the MQSUB call so that it matches the existing subscription.

2512 (09D0) (RC2512): MQRC_SUBLEVEL_NOT_ALTERABLE

Explanation

An MQSUB call using option MQSO_ALTER was made changing the `SubLevel` of the subscription. The `SubLevel` of a subscription cannot be changed.

Completion code

MQCC_FAILED

Programmer response

Remove the subscription using MQCLOSE and re-create it with MQSUB with the attributes set correctly, or change the `SubLevel` field used on the MQSUB call so that it matches the existing subscription.

2513 (09D1) (RC2513): MQRC_PROPERTY_NAME_LENGTH_ERR

Explanation

An attempt was made to set, inquire or delete a property with an invalid name. This is for one of the following reasons:

- The `VSLength` field of the property name was set to less than or equal to zero.
- The `VSLength` field of the property name was set to greater than the maximum allowed value (see constant MQ_MAX_PROPERTY_NAME_LENGTH).
- The `VSLength` field of the property name was set to MQVS_NULL_TERMINATED and the property name was greater than the maximum allowed value.

Completion code

MQCC_FAILED

Programmer response

Valid property names are described in the IBM MQ documentation. Ensure that the property has a valid name length before issuing the call again.

2514 (09D2) (RC2514): MQRC_DUPLICATE_GROUP_SUB

Explanation

An MQSUB call using option MQSO_GROUP_SUB was made creating a new grouped subscription but, although it has a unique SubName, it matches the Full topic name of an existing subscription in the group.

Completion code

MQCC_FAILED

Programmer response

Correct the Full topic name used so that it does not match any existing subscription in the group, or correct the grouping attributes if, either a different group was intended or the subscription was not intended to be grouped at all.

2515 (09D3) (RC2515): MQRC_GROUPING_NOT_ALTERABLE

Explanation

An MQSUB call was made using option MQSO_ALTER on a grouped subscription, that is one made with the option MQSO_GROUP_SUB. Grouping of subscriptions is not alterable.

Completion code

MQCC_FAILED

Programmer response

Remove the subscription using MQCLOSE and re-create it with MQSUB with the attributes set correctly, or change the various grouping fields used on the MQSUB call so that it matches the existing subscription.

2516 (09D4) (RC2516): MQRC_SELECTOR_INVALID_FOR_TYPE

Explanation

A SelectionString may only be specified in the MQOD for an MQOPEN/MQPUT1 if the following is true:

- ObjectType is MQOT_Q
- The queue is being opened using one of the MQOO_INPUT_* open options.

Completion code

MQCC_FAILED

Programmer response

Modify the value of ObjectType to be MQOT_Q and ensure that the queue is being opened using one of the MQOO_INPUT_* options.

2517 (09D5) (RC2517): MQRC_HOBJ_QUIESCED

Explanation

The HOBJ has been quiesced but there are no messages in the read ahead buffer which match the current selection criteria. This reason code indicates that the read ahead buffer is not empty.

Completion code

MQCC_FAILED

Programmer response

This reason code indicates that all messages with the current selection criteria have been processed. Do one of the following:

- If no further messages need to be processed issue an MQCLOSE without the MQCO_QUIESCE option. Any messages in the read ahead buffer will be discarded.
- Relax the current selection criteria by modifying the values in the MQGMO and reissue the call. Once all messages have been consumed the call will return MQRC_HOBJ_QUIESCED_NO_MSGS.

2518 (09D6) (RC2518): MQRC_HOBJ_QUIESCED_NO_MSGS

Explanation

The HOBJ has been quiesced and the read ahead buffer is now empty. No further messages will be delivered to this HOBJ

Completion code

MQCC_FAILED

Programmer response

Issue MQCLOSE against the HOBJ.

2519 (09D7) (RC2519): MQRC_SELECTION_STRING_ERROR

Explanation

The `SelectionString` must be specified according to the description of how to use an MQCHARV structure. Examples of why this error was returned:

- `SelectionString.VSLength` is greater than zero, but `SelectionString.VSOffset` is zero and `SelectionString.VSPtr` is a null pointer.
- `SelectionString.VSOffset` is nonzero and `SelectionString.VSPtr` is not the null pointer (that is, it appears both fields are being used where only one is allowed).
- `SelectionString.VSPtr` is not a valid pointer.
- `SelectionString.VSOffset` or `SelectionString.VSPtr` points to storage that is not accessible.
- `SelectionString.VSLength` exceeds the maximum length allowed for this field. The maximum length is determined by MQ_SELECTOR_LENGTH.

Completion code

MQCC_FAILED

Programmer response

Modify the fields of the MQCHARV so that it follows the rules for a valid MQCHARV structure.

2520 (09D8) (RC2520): MQRC_RES_OBJECT_STRING_ERROR

Explanation

On the MQOPEN or MQPUT1 call in the Object Descriptor MQOD, or on the MQSUB call in the Subscription Descriptor MQSD the `ResObjectString` field is not valid.

One of the following applies:

- `ResObjectString.VSLength` is greater than zero, but `ResObjectString.VSOffset` is zero and `ResObjectString.VSPtr` is the null pointer.
- `ResObjectString.VSOffset` is nonzero and `ResObjectString.VSPtr` is not the null pointer (that is, it appears both fields are being used where only one is allowed).
- `ResObjectString.VSPtr` is not a valid pointer.
- `ResObjectString.VSOffset` or `ResObjectString.VSPtr` points to storage that is not accessible.
- `ResObjectString.VSBufSize` is MQVS_USE_VSLENGTH and one of `ResObjectString.VSOffset` or `ResObjectString.VSPtr` have been provided.

Completion code

MQCC_FAILED

Programmer response

Ensure that one of `ResObjectString.VSOffset` or `ResObjectString.VSPtr` is zero and the other nonzero and that the buffer length is provided in `ResObjectString.VSBufSize`. Ensure that the field used points to accessible storage.

2521 (09D9) (RC2521): MQRC_CONNECTION_SUSPENDED

Explanation

An MQCTL call with Operation MQOP_START_WAIT has returned because the asynchronous consumption of messages has been suspended. This can be for the following reasons:

- The connection was explicitly suspended using MQCTL with Operation MQOP_SUSPEND
- All consumers have been either unregistered or suspended.

Completion code

MQCC_WARNING

Programmer response

If this is an expected condition, no corrective action required. If this is an unexpected condition check that:

- At least one consumer is registered and not suspended
- The connection has not been suspended

2522 (09DA) (RC2522): MQRC_INVALID_DESTINATION

Explanation

An MQSUB call failed because of a problem with the destination where publications messages are to be sent, so an object handle cannot be returned to the application and the subscription is not made. This can be for one of the following reasons:

- The MQSUB call used MQSO_CREATE, MQSO_MANAGED and MQSO_NON_DURABLE and the model queue referred to by MNDURMDL on the administrative topic node does not exist
- The MQSUB call used MQSO_CREATE, MQSO_MANAGED and MQSO_DURABLE and the model queue referred to by MDURMDL on the administrative topic node does not exist, or has been defined with a DEFTYPE of TEMPDYN.
- The MQSUB call used MQSO_CREATE or MQSO_ALTER on a durable subscription and the object handle provided referred to a temporary dynamic queue. This is not an appropriate destination for a durable subscription.
- The MQSUB call used MQSO_RESUME and a Hobj of MQHO_NONE, to resume an administratively created subscription, but the queue name provided in the DEST parameter of the subscription does not exist.
- The MQSUB call used MQSO_RESUME and a Hobj of MQHO_NONE, to resume a previously created API subscription, but the queue previously used no longer exists.

Completion code

MQCC_FAILED

Programmer response

Ensure that the model queues referred to by MNDURMDL and MDURMDL exist and have an appropriate DEFTYPE. Create the queue referred to by the DEST parameter in an administrative subscription if one is being used. Alter the subscription to use an existing queue if the previously used one does not exist.

2523 (09DB) (RC2523): MQRC_INVALID_SUBSCRIPTION

Explanation

An MQSUB call using MQSO_RESUME or MQSO_ALTER failed because the subscription named is not valid for use by applications. This can be for one of the following reasons:

- The subscription is the SYSTEM.DEFAULT.SUB subscription, which is not a valid subscription and should only be used to specify the default values on **DEFINE SUB** commands.
- The subscription is a proxy type subscription, which is not a valid subscription for an application to resume and is only used to enable publications to be forwarded between queue managers.
- The subscription has expired and is no longer valid for use.
- The subscription is a JMS 2.0 shared subscription, which is not a valid subscription for a non-JMS 2.0 application to resume or alter.

Completion code

MQCC_FAILED

Programmer response

Ensure the subscription named in the SubName field is not one of the invalid ones listed. If you have a handle open to the subscription already it must have expired. Use MQCLOSE to close the handle and then if necessary create a new subscription.

2524 (09DC) (RC2524): MQRC_SELECTOR_NOT_ALTERABLE

Explanation

An MQSUB call was issued with the MQSO_ALTER option and the MQSD contained a SelectionString. It is not valid to alter the SelectionString of a subscription.

Completion code

MQCC_FAILED

Programmer response

Ensure that the SelectionString field of the MQSD does not contain a valid VSPtr and that the VSLength is set to zero when making a call to MQSUB.

2525 (09DD) (RC2525): MQRC_RETAINED_MSG_Q_ERROR

Explanation

An MQSUB call which did not use the MQSO_NEW_PUBLICATIONS_ONLY option, or an MQSUBRQ call, failed because the retained publications which exist for the topic string subscribed to cannot be retrieved from the SYSTEM.RETAINED.PUB.QUEUE. This can be for one of the following reasons:

- The queue has become damaged or has been deleted.
- The queue has been set to GET(DISABLED).
- Messages have been removed from this queue directly.

An error message will be written to the log giving more details about the problem with the SYSTEM.RETAINED.PUB.QUEUE.

When this return code occurs on an MQSUB call, it can only occur using the MQSO_CREATE option, and in this case the subscription is not created.

Completion code

MQCC_FAILED

Programmer response

If this occurs on an MQSUB call, re-issue the MQSUB call using the option MQSO_NEW_PUBLICATIONS_ONLY, which will mean no previously retained publications are sent to this subscription, or fix the SYSTEM.RETAINED.PUB.QUEUE so that messages can be retrieved from it and re-issue the MQSUB call.

If this occurs on an MQSUBRQ call, fix the SYSTEM.RETAINED.PUB.QUEUE so that messages can be retrieved from it and re-issue the MQSUBRQ call.

2526 (09DE) (RC2526): MQRC_RETAINED_NOT_DELIVERED

Explanation

An MQSUB call which did not use the MQSO_NEW_PUBLICATIONS_ONLY option or an MQSUBRQ call, failed because the retained publications which exist for the topic string subscribed to cannot be delivered to the subscription destination queue and have subsequently failed to be delivered to the dead-letter queue.

When this return code occurs on an MQSUB call, it can only occur using the MQSO_CREATE option, and in this case the subscription is not created.

Completion code

MQCC_FAILED

Programmer response

Fix the problems with the destination queue and the dead-letter queue and re-issue the MQSUB or MQSUBRQ call.

2527 (09DF) (RC2527): MQRC_RFH_RESTRICTED_FORMAT_ERR

Explanation

A message was put to a queue containing an MQRFH2 header which included a folder with a restricted format. However, the folder was not in the required format. These restrictions are:

- If `NameValueCCSID` of the folder is 1208 then only single byte UTF-8 characters are allowed in the folder, group or element names.
- Groups are not allowed in the folder.
- The values of properties may not contain any characters that require escaping.
- Only Unicode character U+0020 will be treated as white space within the folder.
- The folder tag does not contain the content attribute.
- The folder must not contain a property with a null value.

The *mq* folder requires formatting of this restricted form.

Completion code

MQCC_FAILED

Programmer response

Change the message to include valid MQRFH2 folders.

2528 (09E0) (RC2528): MQRC_CONNECTION_STOPPED

Explanation

An MQCTL call was issued to start the asynchronous consumption of messages, but before the connection was ready to consume messages it was stopped by one of the message consumers.

Completion code

MQCC_FAILED

Programmer response

If this is an expected condition, no corrective action required. If this is an unexpected condition check whether an MQCTL with Operation MQOP_STOP was issued during the MQCBCT_START callback function.

2529 (09E1) (RC2529): MQRC_ASYNC_UOW_CONFLICT

Explanation

An MQCTL call with Operation MQOP_START was issued to start the asynchronous consumption of messages, but the connection handle used already has a global unit of work outstanding. MQCTL cannot be used to start asynchronous consumption of messages while a unit of work is in existence unless the MQOP_START_WAIT Operation is used

Completion code

MQCC_FAILED

Programmer response

Issue an MQCMIT on the connection handle to commit the unit of work and then reissue the MQCTL call, or issue an MQCTL call using Operation MQOP_START_WAIT to use the unit of work from within the asynchronous consumption callback functions.

2530 (09E2) (RC2530): MQRC_ASYNC_XA_CONFLICT

Explanation

An MQCTL call with Operation MQOP_START was issued to start the asynchronous consumption of messages, but an external XA syncpoint coordinator has already issued an xa_open call for this connection handle. XA transactions must be done using the MQOP_START_WAIT Operation.

Completion code

MQCC_FAILED

Programmer response

Reissue the MQCTL call using Operation MQOP_START_WAIT.

2531 (09E3) (RC2531): MQRC_PUBSUB_INHIBITED

Explanation

MQSUB, MQOPEN, MQPUT, and MQPUT1 calls are currently inhibited for all publish/subscribe topics, either with the queue manager attribute PSMODE or because processing of publish/subscribe state at queue manager start-up has failed, or has not yet completed.

Completion code

MQCC_FAILED

Programmer response

If this queue manager does not intentionally inhibit publish/subscribe, investigate any error messages that describe the failure at queue manager start-up, or wait until start-up processing completes. If the

queue manager is a member of cluster, then start-up is not complete until the channel initiator has also started.

z/OS On z/OS, if you get this return code from the Chinit for the SYSTEM.BROKER.DEFAULT.STREAM queue or topic, then the Chinit is busy processing work, and the pubsub task starts later. Use the DISPLAY PUBSUB command to check the status of the publish/subscribe engine to ensure that it is ready for use. Additionally, on z/OS, you might receive an information message CSQM076I.

2532 (09E4) (RC2532): MQRC_MSG_HANDLE_COPY_FAILURE

Explanation

An MQGET call was issued specifying a valid `MsgHandle` in which to retrieve any properties of the message. After the message had been removed from the queue the application could not allocate enough storage for the properties of the message. The message data is available to the application but the properties are not. Check the queue manager error logs for more information about how much storage was required.

Completion code

MQCC_WARNING

Programmer response

Raise the memory limit of the application to allow it store the properties.

2533 (09E5) (RC2533): MQRC_DEST_CLASS_NOT_ALTERABLE

Explanation

An MQSUB call using option MQSO_ALTER was made changing the use of the MQSO_MANAGED option on the subscription. The destination class of a subscription cannot be changed. When the MQSO_MANAGED option is not used, the queue provided can be changed, but the class of destination (managed or not) cannot be changed.

Completion code

MQCC_FAILED

Programmer response

Remove the subscription using MQCLOSE and re-create it with MQSUB with the attributes set correctly, or change the use of the MQSO_MANAGED option used on the MQSUB call so that it matches the existing subscription.

2534 (09E6) (RC2534): MQRC_OPERATION_NOT_ALLOWED

Explanation

An MQCTL call was made with an Operation that is not allowed, because of the current state of asynchronous consumption on the hConn.

If Operation was MQOP_RESUME, the operation is not allowed because the state of asynchronous consumption on the hConn is STOPPED. Re-issue MQCTL with the MQOP_START Operation.

If Operation was MQOP_SUSPEND, the operation is not allowed because the state of asynchronous consumption on the hConn is STOPPED. If you need to get your hConn into a SUSPENDED state, issue MQCTL with the MQOP_START Operation followed by MQCTL with MQOP_SUSPEND.

If Operation was MQOP_START, the operation is not allowed because the state of asynchronous consumption on the hConn is SUSPENDED. Re-issue MQCTL with the MQOP_RESUME Operation.

If Operation was MQOP_START_WAIT, the operation is not allowed because either:

- The state of asynchronous consumption on the hConn is SUSPENDED. Re-issue MQCTL with the MQOP_RESUME Operation.
- The state of asynchronous consumption on the hConn is already STARTED. Do not mix the use of MQOP_START and MQOP_START_WAIT within one application.

Completion code

MQCC_FAILED

Programmer response

Re-issue the MQCTL call with the correct Operation.

2535 (09E7): MQRC_ACTION_ERROR

Explanation

An MQPUT call was issued, but the value of the `Action` field in the **PutMsgOpts** parameter is not a valid MQACTP_* value.

Completion code

MQCC_FAILED

Programmer response

Specify a valid value for the field.

2537 (09E9) (RC2537): MQRC_CHANNEL_NOT_AVAILABLE

Explanation

An MQCONN call was issued from a client to connect to a queue manager but the channel is not currently available. Common causes of this reason code are:

- The channel is currently in stopped state.
- The channel has been stopped by a channel exit.
- The queue manager has reached its maximum allowable limit for this channel from this client.
- The queue manager has reached its maximum allowable limit for this channel.
- The queue manager has reached its maximum allowable limit for all channels

Completion code

MQCC_FAILED

Programmer response

Examine the queue manager and client error logs for messages explaining the cause of the problem.

This reason code is also used to identify the corresponding event message “Channel Not Available” on page 160.

2538 (09EA) (RC2538): MQRC_HOST_NOT_AVAILABLE

Explanation

An MQCONN call was issued from a client to connect to a queue manager but the attempt to allocate a conversation to the remote system failed. Common causes of this reason code are:

- The listener has not been started on the remote system.
- The connection name in the client channel definition is incorrect.
- The network is currently unavailable.
- A firewall blocking the port, or protocol-specific traffic.
- The security call initializing the IBM MQ client is blocked by a security exit on the SVRCONN channel at the server.

Completion code

MQCC_FAILED

Programmer response

Examine the client error log for messages explaining the cause of the problem.

Linux If you are using a Linux server, and receiving a 2538 return code when trying to connect to a queue manager, ensure that you check your internal firewall configuration.

- To diagnose the problem, issue the following commands to temporarily turn off the internal Linux firewall :

```
/etc/init.d/iptables save
/etc/init.d/iptables stop
```

- To turn the internal Linux firewall back on, issue the command:

```
/etc/init.d/iptables start
```

- To permanently turn off the internal Linux firewall, issue the command:

```
chkconfig iptables off
```

2539 (09EB) (RC2539): MQRC_CHANNEL_CONFIG_ERROR

Explanation

An MQCONN call was issued from a client to connect to a queue manager but the attempt to establish communication failed. Common causes of this reason code are:

- The server and client cannot agree on the channel attributes to use.
- There are errors in one or both of the `qm.ini` or `mqclient.ini` configuration files.
- The server machine does not support the code page used by the client.

Completion code

MQCC_FAILED

Programmer response

Examine the queue manager and client error logs for messages explaining the cause of the problem.

2540 (09EC) (RC2540): MQRC_UNKNOWN_CHANNEL_NAME

Explanation

An MQCONN call was issued from a client to connect to a queue manager but the attempt to establish communication failed because the queue manager did not recognize the channel name.

Completion code

MQCC_FAILED

Programmer response

Ensure that the client is configured to use the correct channel name.

2541 (09ED) (RC2541): MQRC_LOOPING_PUBLICATION

Explanation

A distributed Pub/Sub topology has been configured with a combination of Pub/Sub clusters and Pub/Sub Hierarchies such that some, or all, of the queue managers have been connected in a loop. A looping publication has been detected and put onto the dead-letter queue.

Completion code

MQCC_FAILED

Programmer response

Examine the hierarchy and correct the loop.

2543 (09EF) (RC2543): MQRC_STANDBY_Q_MGR

Explanation

The application attempted to connect to a standby queue manager instance.

Standby queue manager instances do not accept connections. To connect to the queue manager, you must connect to its active instance.

Completion code

MQCC_FAILED

Programmer response

Connect the application to an active queue manager instance.

2544 (09F0) (RC2544): MQRC_RECONNECTING

Explanation

The connection has started reconnecting.

If an event handler has been registered with a reconnecting connection, it is called with this reason code when reconnection attempts begin.

Completion code

MQCC_WARNING

Programmer response

Let IBM MQ continue with its next reconnection attempt, change the interval before the reconnection, or stop the reconnection. Change any application state that depends on the reconnection.

Note: Reconnection might start while the application is in the middle of an MQI call.

2545 (09F1) (RC2545): MQRC_RECONNECTED

Explanation

The connection reconnected successfully and all handles are reinstated.

If reconnection succeeds, an event handler registered with the connection is called with this reason code.

Completion code

MQCC_OK

Programmer response

Set any application state that depends on the reconnection.

Note: Reconnection might finish while the application is in the middle of an MQI call.

2546 (09F2) (RC2546): MQRC_RECONNECT_QMID_MISMATCH

Explanation

A reconnectable connection specified MQCNO_RECONNECT_Q_MGR and the connection attempted to reconnect to a different queue manager.

Completion code

MQCC_FAILED

Programmer response

Ensure that the configuration for a reconnectable client resolves to a single queue manager.

If the application does not require reconnection to exactly the same queue manager, use the MQCONNX option MQCNO_RECONNECT.

2547 (09F3) (RC2547): MQRC_RECONNECT_INCOMPATIBLE

Explanation

An MQI option is incompatible with reconnectable connections.

This error indicates that the option relies on information in a queue manager that is lost during reconnection. For example, the option MQPMO_LOGICAL_ORDER, requires the queue manager to remember information about logical message ordering that is lost during reconnection.

This reason also occurs for an MQOPEN when the object being opened is a clustered queue, if the open options include MQOO_BIND_ON_OPEN (or the default bind option for the queue equates to that value), the connection to the queue manager was made using the MQCNO_RECONNECT option, and the queue manager is in a uniform cluster.

You cannot use authentication tokens to connect to IBM MQ if the reconnect option is specified. If an application supplies an authentication token, and specifies the MQCNO_RECONNECT or MQCNO_RECONNECT_Q_MGR option in the MQCNO structure, the connection fails and this reason code is returned to the application.

This reason also occurs if the application instance mode is set to JVM and one of the following statements is true:

- The application attempted to make JMS connections that have incompatible options from within the same client Java virtual machine.
- The application attempted to create connections that are not reconnectable.

For more information about the requirements when using this mode of operation, see JVM mode restrictions.

Completion code

MQCC_FAILED

Programmer response

Modify your application to remove the incompatible option, or do not allow the application to be reconnectable.

2548 (09F4) (RC2548): MQRC_RECONNECT_FAILED

Explanation

After reconnecting, an error occurred while reinstating the handles for a reconnectable connection.

For example, an attempt to reopen a queue that had been open when the connection broke, failed.

Completion code

MQCC_FAILED

Programmer response

Investigate the cause of the error in the error logs. Consider using the MQSTAT API to find further details of the failure.

2549 (09F5) (RC2549): MQRC_CALL_INTERRUPTED

Explanation

MQPUT, MQPUT1, or MQCMIT was interrupted and reconnection processing cannot reestablish a definite outcome.

This reason code is returned to a client that is using a re-connectable connection if the connection is broken between sending the request to the queue manager and receiving the response, and if the outcome is not certain. For example, an interrupted MQPUT of a message outside sync point might or might not have stored the message. Alternatively an interrupted MQPUT1 of a message outside sync point might or might not have stored the message. The timing of the failure affects whether the message remains on the queue or not. If MQCMIT was interrupted the transaction might or might not have been committed.

Completion code

MQCC_FAILED

Programmer response

Repeat the call following reconnection, but be aware that in some cases, repeating the call might be misleading.

The application design determines the appropriate recovery action. In many cases, getting and putting messages inside sync point resolves indeterminate outcomes. Where messages need to be processed outside sync point, it might be necessary to establish whether the interrupted operation succeeded before the interruption and repeating it if it did not.

2550 (09F6) (RC2550): MQRC_NO_SUBS_MATCHED

Explanation

An MQPUT or MQPUT1 call was successful but no subscriptions matched the topic.

Completion code

MQCC_WARNING

Programmer response

No response is required, unless this reason code was not expected by the application that put the message.

2551 (09F7) (RC2551): MQRC_SELECTION_NOT_AVAILABLE

Explanation

An MQSUB call subscribed to publications using a `SelectionString`. IBM MQ is unable to accept the call because it does not follow the rules for specifying selection strings, which are documented in Message selector syntax. It is possible that the selection string is acceptable to an extended message selection provider, however no extended message selection provider was available to validate the selection string. If a subscription is being created, the MQSUB fails; otherwise MQSUB completes with a warning.

An MQPUT or MQPUT1 call published a message and at least one subscriber had a content filter but IBM MQ could not determine whether the publication should be delivered to the subscriber (for example,

because no extended message selection provider was available to validate the selection string). The MQPUT or MQPUT1 call will fail with MQRC_SELECTION_NOT_AVAILABLE and no subscribers will receive the publication.

Completion code

MQCC_WARNING or MQCC_FAILED

Programmer response

If it was intended that the selection string should be handled by the extended message selection provider, ensure that the extended message selection provider is correctly configured and running. If extended message selection was not intended, see Message selector syntax and ensure that you have correctly followed the rules for specifying selection strings.

If an administratively defined subscription selector is using extended message syntax, this is indicated by the **DISPLAY SUB** parameter **SELTYPE** having the value EXTENDED. You can use the following command to identify the administratively defined subscribers that are being interpreted as extended message selectors:

```
DISPLAY SUB(*) WHERE(SELTYPE EQ EXTENDED)
```

If a subscription is being resumed, the subscription will not be delivered any messages until a extended message selection provider is available and a message matches the `SelectionString` of the resumed subscription.

2552 (09F8) (RC2552): MQRC_CHANNEL_SSL_WARNING

Explanation

An TLS security event has occurred. This is not fatal to an TLS connection but is likely to be of interest to an administrator.

Completion code

MQCC_WARNING

Programmer response

None. This reason code is only used to identify the corresponding event message “Channel SSL Warning” on page 164.

2553 (09F9) (RC2553): MQRC_OCSP_URL_ERROR

Explanation

The `OCSPResponderURL` field does not contain a correctly formatted HTTP URL.

Completion code

MQCC_FAILED

Programmer response

Check and correct the `OCSPResponderURL`. If you do not intend to access an OCSP responder, set the **AuthInfoType** of the authentication information object to MQAIT_CRL_LDAP.

2554 (09FA) (RC2554): MQRC_CONTENT_ERROR

Explanation

There are two explanations for reason code 2554:

1. An MQPUT call was issued with a message where the content could not be parsed to determine whether the message should be delivered to a subscriber with an extended message selector. No subscribers will receive the publication.
2. MQRC_CONTENT_ERROR can be returned from MQSUB and MQSUBRQ if a selection string selecting on the content of the message was specified.

Completion code

MQCC_FAILED

Programmer response

There are two programmer responses for reason code 2554 because there are two causes:

1. If reason code 2554 was issued because of reason “1” on page 1342 then check for error messages from the extended message selection provider and ensure that the message content is well formed before retrying the operation.
2. If reason code 2554 was issued because of reason “2” on page 1342 then because the error occurred at the time that the retained message was published, either a system administrator must clear the retained queue, or you cannot specify a selection string selecting on the content.

2555 (09FB) (RC2555): MQRC_RECONNECT_Q_MGR_REQD

Explanation

The MQCNO_RECONNECT_Q_MGR option is required.

An option, such MQMO_MATCH_MSG_TOKEN in an MQGET call or opening a durable subscription, was specified in the client program that requires re-connection to the same queue manager.

Completion code

MQCC_FAILED

Programmer response

Change the MQCONNX call to use MQCNO_RECONNECT_Q_MGR, or modify the client program not to use the conflicting option.

2556 (09FC) (RC2556): MQRC_RECONNECT_TIMED_OUT

Explanation

A reconnection attempt timed out.

The failure might occur in any MQI verb if a connection is configured to reconnect. You can customize the timeout in the `MQClient.ini` file

Completion code

MQCC_FAILED

Programmer response

Look at the error logs to find out why reconnection did not complete within the time limit.

Multi 2557 (09FD) (RC2557): MQRC_PUBLISH_EXIT_ERROR

Explanation

A publish exit function returned an invalid response code, or failed in some other way. This can be returned from the MQPUT, MQPUT1, MQSUB and MQSUBRQ function calls.

Completion code

MQCC_FAILED

Programmer response

Check the publish exit logic to ensure that the exit is returning valid values in the `ExitResponse` field of the MQPSXP structure. Consult the IBM MQ error log files and FFST records for more details about the problem.

2558 (09FE) (RC2558): MQRC_COMMINFO_ERROR

Explanation

The configuration of either the name of the COMMINFO object or the object itself is incorrect.

Completion code

MQCC_FAILED

Programmer response

Check the configuration of the TOPIC and COMMINFO objects and retry the operation.

2560 (0A00) (RC2560): MQRC_MULTICAST_ONLY

Explanation

An attempt was made to use a topic which is defined as multicast only in a non-multicast way. Possible causes for this error are:

1. An MQPUT1 call was issued to the topic
2. An MQOPEN call was issued using the MQOO_NO_MULTICAST option
3. An MQSUB call was issued using the MQSO_NO_MULTICAST option
4. The application is connected directly through bindings, that is, there is no client connection
5. The application is being run from a release prior to IBM WebSphere MQ 7.1

Completion code

MQCC_FAILED

Programmer response

Either change the topic definition to enable non-multicast, or change the application.

z/OS 2561 (0A01) (RC2561): MQRC_DATA_SET_NOT_AVAILABLE

Explanation

An IBM MQ MQI call or command was issued to operate on a shared queue, but the call failed because the data for the shared message has been offloaded to a shared message data set that is temporarily unavailable to the current queue manager. This can occur either because of a problem in accessing the data set or because the data set was previously found to be damaged, and is awaiting completion of recovery processing.

This return code can also occur if the shared message data set has not been defined for the queue manager being used. You might be using the wrong queue manager in the queue sharing group.

- This reason code occurs only on z/OS.

Completion code

MQCC_FAILED

Programmer response

The problem is temporary; wait a short while, and then retry the operation.

Use `DIS CFSTRUCT(...) SMDSCONN(*)` to display the status of the SMDS connection.

To start the connection if the STATUS is not OPEN, use `STA SMDSCONN(*) CFSTRUCT(...)`.

Use `DISPLAY CFSTATUS(...) TYPE(SMDS)` and check the status is active on the queue manager that you are using.

2562 (0A02) (RC2562): MQRC_GROUPING_NOT_ALLOWED

Explanation

An MQPUT call was issued to put a grouped message to a handle which is publishing over multicast.

Completion code

MQCC_FAILED

Programmer response

Either change the topic definition to disable multicast or change the application to not use grouped messages.

2563 (0A03) (RC2563): MQRC_GROUP_ADDRESS_ERROR

Explanation

An MQOPEN or MQSUB call was issued to a multicast topic which has been defined with an incorrect group address field.

Completion code

MQCC_FAILED

Programmer response

Correct the group address field in the COMMINFO definition linked to the TOPIC object.

2564 (0A04) (RC2564): MQRC_MULTICAST_CONFIG_ERROR

Explanation

An MQOPEN, MQSUB or MQPUT call was issued which invoked the multicast component. The call failed because the multicast configuration is incorrect.

Completion code

MQCC_FAILED

Programmer response

Check the multicast configuration and error logs and retry the operation.

2565 (0A05) (RC2565): MQRC_MULTICAST_INTERFACE_ERROR

Explanation

An MQOPEN, MQSUB or MQPUT call was made which attempted to a network interface for multicast. The interface returned an error. Possible causes for the error are:

1. The required network interface does not exist.
2. The interface is not active.
3. The interface does not support the required IP version.

Completion code

MQCC_FAILED

Programmer response

Verify that the IP address and the system network configuration are valid. Check the multicast configuration and error logs and retry the operation.

2566 (0A06) (RC2566): MQRC_MULTICAST_SEND_ERROR

Explanation

An MQPUT call was made which attempted to send multicast traffic over the network. The system failed to send one or more network packets.

Completion code

MQCC_FAILED

Programmer response

Verify that the IP address and the system network configuration are valid. Check the multicast configuration and error logs and retry the operation.

2567 (0A07) (RC2567): MQRC_MULTICAST_INTERNAL_ERROR

Explanation

An MQOPEN, MQSUB or MQPUT call was issued which invoked the multicast component. An internal error occurred which prevented the operation completing successfully.

Completion code

MQCC_FAILED

Programmer response

Tell the systems administrator.

2568 (0A08) (RC2568): MQRC_CONNECTION_NOT_AVAILABLE

Explanation

An MQCONN or MQCONNX call was made when the queue manager was unable to provide a connection of the requested connection type on the current installation. A client connection cannot be made on a server only installation. A local connection cannot be made on a client only installation.

This error can also occur when IBM MQ fails an attempt to load a library from the installation that the requested queue manager is associated with.

Completion code

MQCC_FAILED

Programmer response

Ensure that the connection type requested is applicable to the type of installation. If the connection type is applicable to the installation then consult the error log for more information about the nature of the error.

2569 (0A09) (RC2569): MQRC_SYNCPOINT_NOT_ALLOWED

Explanation

An MQPUT or MQPUT1 call using MQPMO_SYNCPOINT was made to a topic that is defined as **MCAST(ENABLED)**. This is not allowed.

Completion code

MQCC_FAILED

Programmer response

Change the application to use MQPMO_NO_SYNCPOINT, or alter the topic to disable the use of Multicast and retry the operation.

2577 (0A11) (RC2577): MQRC_CHANNEL_BLOCKED

Explanation

An inbound channel attempted to connect to the queue manager but was blocked due to matching a Channel Authentication rule.

Completion code

MQCC_WARNING

Programmer response

None. This reason code is only used to identify the corresponding event message "Channel Blocked" on page 154.

2578 (0A12) (RC2578): MQRC_CHANNEL_BLOCKED_WARNING

Explanation

An inbound channel attempted to connect to the queue manager and would have been blocked due to matching a Channel Authentication rule, however the rule was defined with **WARN(YES)** so the rule did not block the connection.

Completion code

MQCC_WARNING

Programmer response

None. This reason code is only used to identify the corresponding event message "Channel Blocked" on page 154.

2583 (0A17) (RC2583): MQRC_INSTALLATION_MISMATCH

Explanation

The application attempted to connect to a queue manager that is not associated with the same IBM MQ installation as the loaded libraries.

Completion code

MQCC_FAILED

Programmer response

An application must use the libraries from the installation the queue manager is associated with. If the **AMQ_SINGLE_INSTALLATION** environment variable is set, you must ensure that the application connects only to queue managers associated with a single installation. Otherwise, if IBM MQ is unable to automatically locate the correct libraries, you must modify the application, or the library search path, to ensure that the correct libraries are used.

2587 (0A1B) (RC2587): MQRC_HMSG_NOT_AVAILABLE

Explanation

On an MQGET, MQPUT, or MQPUT1 call, a message handle supplied is not valid with the installation the queue manager is associated with. The message handle was created by MQCRTMH specifying the MQHC_UNASSOCIATED_HCONN option. It can be used only with queue managers associated with the first installation used in the process.

Completion code

MQCC_FAILED

Programmer response

To pass properties between two queue managers associated with different installations, convert the message handle retrieved using MQGET into a buffer using the MQMHBUF call. Then pass that buffer into the MQPUT or MQPUT1 call of the other queue manager. Alternatively, use the **setmqm** command to associate one of the queue managers with the installation that the other queue manager is using. Using the **setmqm** command might change the version of IBM MQ that the queue manager uses.

2589 (0A1D) (RC2589) MQRC_INSTALLATION_MISSING

Explanation

On an MQCONN or MQCONNX call, an attempt was made to connect to a queue manager where the associated installation is no longer installed.

Completion code

MQCC_FAILED

Programmer response

Associate the queue manager with a different installation using the **setmqm** command before attempting to connect to the queue manager again.

2590 (0A1E) (RC2590): MQRC_FASTPATH_NOT_AVAILABLE

Explanation

On an MQCONNX call, the MQCNO_FASTPATH_BINDING option was specified. However, a fastpath connection to the queue manager cannot be made. This issue can occur when a non-fastpath connection to a queue manager was made in the process before this MQCONNX call.

Completion code

MQCC_FAILED

Programmer response

Either change all MQCONNX calls within the process to be fastpath, or use the **AMQ_SINGLE_INSTALLATION** environment variable to restrict connections to a single installation, allowing the queue manager to accept fastpath and non-fastpath connections from the same process, in any order.

2591 (0A1F) (RC2591): MQRC_CIPHER_SPEC_NOT_SUITE_B

Explanation

A client application is configured for NSA Suite B compliant operation but the CipherSpec for the client connection channel is not permitted at the configured Suite B security level. This can occur for Suite B CipherSpecs which fall outside the currently configured security level, for example if ECDHE_ECDSA_AES_128_GCM_SHA256, which is 128-bit Suite B, is used when only the 192-bit Suite B security level is configured.

For more information about which CipherSpecs are Suite B compliant, refer to Specifying CipherSpecs.

Completion code

MQCC_FAILED

Programmer response

Select an appropriate CipherSpec which is permitted at the configured Suite B security level.

2592 (0A20) (RC2592): MQRC_SUITE_B_ERROR

Explanation

The configuration of Suite B is invalid. For example, an unrecognized value was specified in the **MQSUITEB** environment variable, the `EncryptionPolicySuiteB` TLS stanza setting or the MQSCO `EncryptionPolicySuiteB` field.

Completion code

MQCC_FAILED

Programmer response

Determine the fault in the Suite B configuration and amend.

2593 (0A21)(RC2593): MQRC_CERT_VAL_POLICY_ERROR

Explanation

The certificate validation policy configuration is invalid. An unrecognized or unsupported value was specified in the **MQCERTVPOL** environment variable, the `CertificateValPolicy` TLS stanza setting or the MQSCO `CertificateValPolicy` field.

Completion code

MQCC_FAILED

Programmer response

Specify a valid certificate validation policy which is supported on the current platform.

2594 (0A22)(RC2594): MQRC_PASSWORD_PROTECTION_ERROR

Explanation

An MQCONN or MQCONNX call was issued by a client application, but it failed to agree a password protection algorithm with the queue manager. For unencrypted channels, clients try to agree a password protection mechanism to avoid sending application credentials in plain text across a network.

The usual cause of this error is that the **PasswordProtection** attribute in the **Channels** stanza of the `mqclient.ini` or `qm.ini` file is set to ALWAYS, but the version of IBM MQ that is installed on the remote system does not support password protection.

If the connection passes through IBM MQ Internet Pass-Thru (MQIPT), and the MQIPT route is configured to add or remove TLS encryption, the client and the queue manager might fail to agree a password protection algorithm because one end of the channel is using TLS encryption, and the other is not.

Java and JMS clients must enable MQCSP authentication mode in order to use the **PasswordProtection** feature. See Connection authentication with the Java client.

Completion code

MQCC_FAILED

Programmer response

Consider changing the **PasswordProtection** attribute, or use TLS to protect application credentials instead. If you are using TLS, you must not use a null cipher. A null cipher provides no protection as it causes credentials to be sent in plain text.

More information can be found in the error log in message AMQ9296E.

If the connection passes through MQIPT, and the MQIPT route is configured to add or remove TLS encryption, consider setting the value of the **PasswordProtection** property in the MQIPT route configuration to `compatible`.

2595 (0A23)(RC2595): MQRC_CSP_ERROR

Explanation

The connect call failed because the MQCSP structure is not valid for one of the following reasons:

- The `StrucId` field is not MQCSP_STRUC_ID
- The `Version` field specifies a value that is not valid or not supported.
- The `AuthenticationType` field specifies a value that is not valid or not supported for this version of the MQCSP structure.
- The user identifier is incorrectly specified.
- The password is incorrectly specified.
- The authentication token is incorrectly specified.
- A user ID or password is specified and the `AuthenticationType` field is set to `MQCSP_AUTH_ID_TOKEN`.
- An authentication token is specified and the `AuthenticationType` field is not set to `MQCSP_AUTH_ID_TOKEN`.
- An authentication token is specified by an application connecting using local bindings.

Completion code

MQCC_FAILED

Programmer response

Ensure that the MQCSP structure is correct.

z/OS On z/OS:

- Check that the IBM MQ libraries in STEPLIB are at the same or a higher level than the queue manager.
- If you are using z/OS UNIX System Services, check that the LIBPATH has matching libraries, for example `LIBPATH=$LIBPATH:"/mqm/V8R0M0/java/lib/"`.

2596 (0A24)(RC2596): MQRC_CERT_LABEL_NOT_ALLOWED

Explanation

The channel definition specifies a certificate label but the environment does not support certificate label configuration.

Completion code

MQCC_FAILED

Programmer response

Either, remove the certificate label from the channel definition, or change the configuration to ignore the label.

2598 (0A26)(RC2598): MQRC_ADMIN_TOPIC_STRING_ERROR

Explanation

This error can occur when calling MQSUB or MQOPEN. Publishing to an IBM MQ administrative topic string, starting `$SYS/MQ/` is not permitted.

When subscribing to an IBM MQ administrative topic string, the use of wildcard characters is restricted. For more information, see System topics for monitoring and activity trace.

Completion code

MQCC_FAILED

Programmer response

Change the configuration to publish to an administrative topic string that does not start `$SYS/MQ/`.

Multi 2601 (0A29)(RC2601): MQRC_Q_MGR_RECONNECT_REQUESTED

Explanation

This code can occur when an explicit reconnection request is issued, for example, when application balancing is in use.

Completion code

MQCC_WARNING

2602 (0A2A)(RC2602): MQRC_BNO_ERROR

Explanation

When MQBNO is provided by the application code:

- Version not valid or incorrect structure ID
- Other fields outside the valid range or constants as documented in the MQBNO structure.

However, you should note also that an MQBNO structure might be created on behalf of the application code from the `mqclient.ini` **`Application`** or **`ApplicationDefault`** stanzas. Therefore, even if the application code does not directly supply an MQBNO structure, missing or incorrect values in the `mqclient.ini` file can result in this return code.

Completion code

MQCC_WARNING

6100 (17D4) (RC6100): MQRC_REOPEN_EXCL_INPUT_ERROR

Explanation

An open object does not have the correct ImqObject **open options** and requires one or more additional options. An implicit reopen is required but closure has been prevented.

Closure has been prevented because the queue is open for exclusive input and closure might result in the queue being accessed by another process or thread, before the queue is reopened by the process or thread that presently has access.

This reason code occurs in the IBM MQ C++ environment.

Completion code

MQCC_FAILED

Programmer response

Set the **open options** explicitly to cover all eventualities so that implicit reopening is not required.

6101 (17D5) (RC6101): MQRC_REOPEN_INQUIRE_ERROR

Explanation

An open object does not have the correct ImqObject **open options** and requires one or more additional options. An implicit reopen is required but closure has been prevented.

Closure has been prevented because one or more characteristics of the object need to be checked dynamically prior to closure, and the **open options** do not already include MQOO_INQUIRE.

This reason code occurs in the IBM MQ C++ environment.

Completion code

MQCC_FAILED

Programmer response

Set the **open options** explicitly to include MQOO_INQUIRE.

6102 (17D6) (RC6102): MQRC_REOPEN_SAVED_CONTEXT_ERR

Explanation

An open object does not have the correct ImqObject **open options** and requires one or more additional options. An implicit reopen is required but closure has been prevented.

Closure has been prevented because the queue is open with MQOO_SAVE_ALL_CONTEXT, and a destructive get has been performed previously. This has caused retained state information to be associated with the open queue and this information would be destroyed by closure.

This reason code occurs in the IBM MQ C++ environment.

Completion code

MQCC_FAILED

Programmer response

Set the **open options** explicitly to cover all eventualities so that implicit reopening is not required.

6103 (17D7) (RC6103): MQRC_REOPEN_TEMPORARY_Q_ERROR

Explanation

An open object does not have the correct ImqObject **open options** and requires one or more additional options. An implicit reopen is required but closure has been prevented.

Closure has been prevented because the queue is a local queue of the definition type MQQDT_TEMPORARY_DYNAMIC, that would be destroyed by closure.

This reason code occurs in the IBM MQ C++ environment.

Completion code

MQCC_FAILED

Programmer response

Set the **open options** explicitly to cover all eventualities so that implicit reopening is not required.

6104 (17D8) (RC6104): MQRC_ATTRIBUTE_LOCKED

Explanation

An attempt has been made to change the value of an attribute of an object while that object is open, or, for an ImqQueueManager object, while that object is connected. Certain attributes cannot be changed in these circumstances. Close or disconnect the object (as appropriate) before changing the attribute value.

An object might have been connected, opened, or both unexpectedly and implicitly to perform an MQINQ call. Check the attribute cross-reference table in C++ and MQI cross-reference to determine whether any of your method invocations result in an MQINQ call.

This reason code occurs in the IBM MQ C++ environment.

Completion code

MQCC_FAILED

Programmer response

Include MQOO_INQUIRE in the ImqObject **open options** and set them earlier.

6105 (17D9) (RC6105): MQRC_CURSOR_NOT_VALID

Explanation

The browse cursor for an open queue has been invalidated since it was last used by an implicit reopen.

This reason code occurs in the IBM MQ C++ environment.

Completion code

MQCC_FAILED

Programmer response

Set the ImqObject **open options** explicitly to cover all eventualities so that implicit reopening is not required.

6106 (17DA) (RC6106): MQRC_ENCODING_ERROR

Explanation

The encoding of the (next) message item needs to be MQENC_NATIVE for pasting.

This reason code occurs in the IBM MQ C++ environment.

Completion code

MQCC_FAILED

6107 (17DB) (RC6107): MQRC_STRUC_ID_ERROR

Explanation

The structure ID for the (next) message item, which is derived from the 4 characters beginning at the data pointer, is either missing or is inconsistent with the class of object into which the item is being pasted.

This reason code occurs in the IBM MQ C++ environment.

Completion code

MQCC_FAILED

6108 (17DC) (RC6108): MQRC_NULL_POINTER

Explanation

A null pointer has been supplied where a nonnull pointer is either required or implied.

This reason code occurs in the IBM MQ C++ environment.

Completion code

MQCC_FAILED

6109 (17DD) (RC6109): MQRC_NO_CONNECTION_REFERENCE

Explanation

The **connection reference** is null. A connection to an ImqQueueManager object is required.

This reason code occurs in the IBM MQ C++ environment.

Completion code

MQCC_FAILED

6110 (17DE) (RC6110): MQRC_NO_BUFFER

Explanation

No buffer is available. For an ImqCache object, one cannot be allocated, denoting an internal inconsistency in the object state that should not occur.

This reason code occurs in the IBM MQ C++ environment.

Completion code

MQCC_FAILED

6111 (17DF) (RC6111): MQRC_BINARY_DATA_LENGTH_ERROR

Explanation

The length of the binary data is inconsistent with the length of the target attribute. Zero is a correct length for all attributes.

- The correct length for an **accounting token** is MQ_ACCOUNTING_TOKEN_LENGTH.
- The correct length for an **alternate security id** is MQ_SECURITY_ID_LENGTH.
- The correct length for a **correlation id** is MQ_CORREL_ID_LENGTH.
- The correct length for a **facility token** is MQ_FACILITY_LENGTH.
- The correct length for a **group id** is MQ_GROUP_ID_LENGTH.
- The correct length for a **message id** is MQ_MSG_ID_LENGTH.
- The correct length for an **instance id** is MQ_OBJECT_INSTANCE_ID_LENGTH.
- The correct length for a **transaction instance id** is MQ_TRAN_INSTANCE_ID_LENGTH.
- The correct length for a **message token** is MQ_MSG_TOKEN_LENGTH.

This reason code occurs in the IBM MQ C++ environment.

Completion code

MQCC_FAILED

6112 (17E0) (RC6112): MQRC_BUFFER_NOT_AUTOMATIC

Explanation

A user-defined (and managed) buffer cannot be resized. A user-defined buffer can only be replaced or withdrawn. A buffer must be automatic (system-managed) before it can be resized.

This reason code occurs in the IBM MQ C++ environment.

Completion code

MQCC_FAILED

Programmer response

6113 (17E1) (RC6113): MQRC_INSUFFICIENT_BUFFER

Explanation

There is insufficient buffer space available after the data pointer to accommodate the request. This might be because the buffer cannot be resized.

This reason code occurs in the IBM MQ C++ environment.

Completion code

MQCC_FAILED

6114 (17E2) (RC6114): MQRC_INSUFFICIENT_DATA

Explanation

There is insufficient data after the data pointer to accommodate the request.

This reason code occurs in the IBM MQ C++ environment.

Completion code

MQCC_FAILED

6115 (17E3) (RC6115): MQRC_DATA_TRUNCATED

Explanation

Data has been truncated when copying from one buffer to another. This might be because the target buffer cannot be resized, or because there is a problem addressing one or other buffer, or because a buffer is being downsized with a smaller replacement.

This reason code occurs in the IBM MQ C++ environment.

Completion code

MQCC_FAILED

6116 (17E4) (RC6116): MQRC_ZERO_LENGTH

Explanation

A zero length has been supplied where a positive length is either required or implied.

This reason code occurs in the IBM MQ C++ environment.

Completion code

MQCC_FAILED

6117 (17E5) (RC6117): MQRC_NEGATIVE_LENGTH

Explanation

A negative length has been supplied where a zero or positive length is required.

This reason code occurs in the IBM MQ C++ environment.

Completion code

MQCC_FAILED

6118 (17E6) (RC6118): MQRC_NEGATIVE_OFFSET

Explanation

A negative offset has been supplied where a zero or positive offset is required.

This reason code occurs in the IBM MQ C++ environment.

Completion code

MQCC_FAILED

6119 (17E7) (RC6119): MQRC_INCONSISTENT_FORMAT

Explanation

The format of the (next) message item is inconsistent with the class of object into which the item is being pasted.

This reason code occurs in the IBM MQ C++ environment.

Completion code

MQCC_FAILED

6120 (17E8) (RC6120): MQRC_INCONSISTENT_OBJECT_STATE

Explanation

There is an inconsistency between this object, which is open, and the referenced ImqQueueManager object, which is not connected.

This reason code occurs in the IBM MQ C++ environment.

Completion code

MQCC_FAILED

6121 (17E9) (RC6121): MQRC_CONTEXT_OBJECT_NOT_VALID

Explanation

The ImqPutMessageOptions **context reference** does not reference a valid ImqQueue object. The object has been previously destroyed.

This reason code occurs in the IBM MQ C++ environment.

Completion code

MQCC_FAILED

6122 (17EA) (RC6122): MQRC_CONTEXT_OPEN_ERROR

Explanation

The ImqPutMessageOptions **context reference** references an ImqQueue object that could not be opened to establish a context. This might be because the ImqQueue object has inappropriate **open options**. Inspect the referenced object **reason code** to establish the cause.

This reason code occurs in the IBM MQ C++ environment.

Completion code

MQCC_FAILED

6123 (17EB) (RC6123): MQRC_STRUC_LENGTH_ERROR

Explanation

The length of a data structure is inconsistent with its content. For an MQRMH, the length is insufficient to contain the fixed fields and all offset data.

This reason code occurs in the IBM MQ C++ environment.

Completion code

MQCC_FAILED

6124 (17EC) (RC6124): MQRC_NOT_CONNECTED

Explanation

A method failed because a required connection to a queue manager was not available, and a connection cannot be established implicitly because the IMQ_IMPL_CONN flag of the ImqQueueManager **behavior** class attribute is FALSE.

This reason code occurs in the IBM MQ C++ environment.

Completion code

MQCC_FAILED

Programmer response

Establish a connection to a queue manager and retry.

6125 (17ED) (RC6125): MQRC_NOT_OPEN

Explanation

A method failed because an object was not open, and opening cannot be accomplished implicitly because the IMQ_IMPL_OPEN flag of the ImqObject **behavior** class attribute is FALSE.

This reason code occurs in the IBM MQ C++ environment.

Completion code

MQCC_FAILED

Programmer response

Open the object and retry.

6126 (17EE) (RC6126): MQRC_DISTRIBUTION_LIST_EMPTY

Explanation

An ImqDistributionList failed to open because there are no ImqQueue objects referenced.

This reason code occurs in the IBM MQ C++ environment.

Completion code

MQCC_FAILED

Programmer response

Establish at least one ImqQueue object in which the **distribution list reference** addresses the ImqDistributionList object, and retry.

6127 (17EF) (RC6127): MQRC_INCONSISTENT_OPEN_OPTIONS

Explanation

A method failed because the object is open, and the ImqObject open options are inconsistent with the required operation. The object cannot be reopened implicitly because the IMQ_IMPL_OPEN flag of the ImqObject **behavior** class attribute is false.

This reason code occurs in the IBM MQ C++ environment.

Completion code

MQCC_FAILED

Programmer response

Open the object with appropriate ImqObject open options and retry.

6128 (17FO) (RC6128): MQRC_WRONG_VERSION

Explanation

A method failed because a version number specified or encountered is either incorrect or not supported.

For the ImqCICSBridgeHeader class, the problem is with the **version** attribute.

This reason code occurs in the IBM MQ C++ environment.

Completion code

MQCC_FAILED

Programmer response

If you are specifying a version number, use one that is supported by the class. If you are receiving message data from another program, ensure that both programs are using consistent and supported version numbers.

6129 (17F1) (RC6129): MQRC_REFERENCE_ERROR

Explanation

An object reference is invalid.

There is a problem with the address of a referenced object. At the time of use, the address of the object is nonnull, but is invalid and cannot be used for its intended purpose.

This reason code occurs in the IBM MQ C++ environment.

Completion code

MQCC_FAILED

Programmer response

Check that the referenced object is neither deleted nor out of scope, or remove the reference by supplying a null address value.

PCF reason codes

Reason codes might be returned by a broker in response to a command message in PCF format, depending on the parameters used in that message.

For more information about PCFs, see Introduction to Programmable Command Formats, Automating administration tasks, and Using Programmable Command Formats.

The following is a list of PCF reason codes, in numeric order, providing detailed information to help you understand them, including:

- An explanation of the circumstances that have caused the code to be raised
- The associated completion code
- Suggested programmer actions in response to the code

Related concepts

“IBM MQ for z/OS messages, completion, and reason codes” on page 289
Use this topic to interpret and understand the messages and codes issued by IBM MQ for z/OS.

Related reference

“AMQ messages on Multiplatforms” on page 244
The AMQ series of diagnostic messages are listed in this section in numerical order, grouped according to the part of IBM MQ from which they originate.

“API completion and reason codes” on page 1151

For each call, a completion code and a reason code are returned by the queue manager or by an exit routine, to indicate the success or failure of the call.

"Transport Layer Security (TLS) return codes" on page 1441
IBM MQ can use TLS with the various communication protocols. Use this topic to identify the error codes that can be returned by TLS.

"WCF custom channel exceptions" on page 1446
Diagnostic messages are listed in this topic in numeric order, grouped according to the part of the Microsoft Windows Communication Foundation (WCF) custom channel from which they originate.

3001 (0BB9) (RC3001): MQRCCF_CFH_TYPE_ERROR

Explanation

Type not valid.

The MQCFH *Type* field value was not valid.

Programmer response

Specify a valid type.

3002 (0BBA) (RC3002): MQRCCF_CFH_LENGTH_ERROR

Explanation

Structure length not valid.

The MQCFH *StrucLength* field value was not valid.

Programmer response

Specify a valid structure length.

3003 (0BBB) (RC3003): MQRCCF_CFH_VERSION_ERROR

Explanation

Structure version number is not valid.

The MQCFH *Version* field value was not valid.

Note that z/OS requires MQCFH_VERSION_3.

Programmer response

Specify a valid structure version number.

3004 (0BBC) (RC3004): MQRCCF_CFH_MSG_SEQ_NUMBER_ERR

Explanation

Message sequence number not valid.

The MQCFH *MsgSeqNumber* field value was not valid.

Programmer response

Specify a valid message sequence number.

3005 (0BBD) (RC3005): MQRCCF_CFH_CONTROL_ERROR

Explanation

Control option not valid.

The MQCFH *Control* field value was not valid.

Programmer response

Specify a valid control option.

3006 (0BBE) (RC3006): MQRCCF_CFH_PARM_COUNT_ERROR

Explanation

Parameter count not valid.

The MQCFH *ParameterCount* field value was not valid.

Programmer response

Specify a valid parameter count.

3007 (0BBF) (RC3007): MQRCCF_CFH_COMMAND_ERROR

Explanation

Command identifier not valid.

The MQCFH *Command* field value was not valid.

Programmer response

Specify a valid command identifier.

3008 (0BC0) (RC3008): MQRCCF_COMMAND_FAILED

Explanation

Command failed.

The command has failed.

Programmer response

Refer to the previous error messages for this command.

3009 (0BC1) (RC3009): MQRCCF_CFIN_LENGTH_ERROR

Explanation

Structure length not valid.

The MQCFIN or MQCFIN64 *StrucLength* field value was not valid.

Programmer response

Specify a valid structure length.

3010 (0BC2) (RC3010): MQRCCF_CFST_LENGTH_ERROR

Explanation

Structure length not valid.

The MQCFST *StrucLength* field value was not valid. The value was not a multiple of four or was inconsistent with the MQCFST *StringLength* field value.

Programmer response

Specify a valid structure length.

3011 (0BC3) (RC3011): MQRCCF_CFST_STRING_LENGTH_ERR

Explanation

String length not valid.

The MQCFST *StringLength* field value was not valid. The value was negative or greater than the maximum permitted length of the parameter specified in the *Parameter* field.

Programmer response

Specify a valid string length for the parameter.

3012 (0BC4) (RC3012): MQRCCF_FORCE_VALUE_ERROR

Explanation

Force value not valid.

The force value specified was not valid.

Programmer response

Specify a valid force value.

3013 (0BC5) (RC3013): MQRCCF_STRUCTURE_TYPE_ERROR

Explanation

Structure type not valid.

The structure *Type* value was not valid.

Programmer response

Specify a valid structure type.

3014 (0BC6) (RC3014): MQRCCF_CFIN_PARM_ID_ERROR

Explanation

Parameter identifier is not valid.

The MQCFIN or MQCFIN64 *Parameter* field value was not valid.

For example, specifying a queue parameter MQIACF_OLDEST_MSG_AGE on in inquire channel status request.

Programmer response

Specify a valid parameter identifier.

3015 (0BC7) (RC3015): MQRCCF_CFST_PARM_ID_ERROR

Explanation

Parameter identifier is not valid.

The MQCFST *Parameter* field value was not valid.

Programmer response

Specify a valid parameter identifier.

3016 (0BC8) (RC3016): MQRCCF_MSG_LENGTH_ERROR

Explanation

Message length not valid.

The message data length was inconsistent with the length implied by the parameters in the message, or a positional parameter was out of sequence.

Programmer response

Specify a valid message length, and check that positional parameters are in the correct sequence.

3017 (0BC9) (RC3017): MQRCCF_CFIN_DUPLICATE_PARM

Explanation

Duplicate parameter.

Two MQCFIN or MQCFIN64 or MQCFIL or MQCFIL64 structures, or any two of those types of structure, with the same parameter identifier were present.

Programmer response

Check for and remove duplicate parameters.

3018 (0BCA) (RC3018): MQRCCF_CFST_DUPLICATE_PARM

Explanation

Duplicate parameter.

Two MQCFST structures, or an MQCFSL followed by an MQCFST structure, with the same parameter identifier were present.

Programmer response

Check for and remove duplicate parameters.

3019 (0BCB) (RC3019): MQRCCF_PARM_COUNT_TOO_SMALL

Explanation

Parameter count too small.

The MQCFH *ParameterCount* field value was less than the minimum required for the command.

Programmer response

Specify a parameter count that is valid for the command.

3020 (0BCC) (RC3020): MQRCCF_PARM_COUNT_TOO_BIG

Explanation

Parameter count too big.

The MQCFH *ParameterCount* field value was more than the maximum for the command.

Programmer response

Specify a parameter count that is valid for the command.

3021 (0BCD) (RC3021): MQRCCF_Q_ALREADY_IN_CELL

Explanation

Queue already exists in cell.

An attempt was made to define a queue with cell scope, or to change the scope of an existing queue from queue manager scope to cell scope, but a queue with that name already existed in the cell.

Programmer response

Do one of the following:

- Delete the existing queue and retry the operation.
- Change the scope of the existing queue from cell to queue manager and retry the operation.
- Create the new queue with a different name.

3022 (0BCE) (RC3022): MQRCCF_Q_TYPE_ERROR

Explanation

Queue type not valid.

The *QType* value was not valid.

Programmer response

Specify a valid queue type.

3023 (0BCF) (RC3023): MQRCCF_MD_FORMAT_ERROR

Explanation

Format not valid.

The MQMD *Format* field value was not MQFMT_ADMIN.

Programmer response

Specify the valid format.

3024 (0BD0) (RC3024): MQRCCF_CFSL_LENGTH_ERROR

Explanation

Structure length not valid.

The MQCFSL *StrucLength* field value was not valid. The value was not a multiple of four or was inconsistent with the MQCFSL *StringLength* field value.

Programmer response

Specify a valid structure length.

3025 (0BD1) (RC3025): MQRCCF_REPLACE_VALUE_ERROR

Explanation

Replace value not valid.

The *Replace* value was not valid.

Programmer response

Specify a valid replace value.

3026 (0BD2) (RC3026): MQRCCF_CFIL_DUPLICATE_VALUE

Explanation

Duplicate parameter value.

In the MQCFIL or MQCFIL64 structure, there was a duplicate parameter value in the list.

Programmer response

Check for and remove duplicate parameter values.

3027 (0BD3) (RC3027): MQRCCF_CFIL_COUNT_ERROR

Explanation

Count of parameter values not valid.

The MQCFIL or MQCFIL64 *Count* field value was not valid. The value was negative or greater than the maximum permitted for the parameter specified in the *Parameter* field.

Programmer response

Specify a valid count for the parameter.

3028 (0BD4) (RC3028): MQRCCF_CFIL_LENGTH_ERROR

Explanation

Structure length not valid.

The MQCFIL or MQCFIL64 *StrucLength* field value was not valid.

Programmer response

Specify a valid structure length.

3029 (0BD5) (RC3029): MQRCCF_MODE_VALUE_ERROR

Explanation

Mode value not valid.

The *Mode* value was not valid.

Programmer response

Specify a valid mode value.

3029 (0BD5) (RC3029): MQRCCF_QUIESCE_VALUE_ERROR

Explanation

Former name for MQRCCF_MODE_VALUE_ERROR.

3030 (0BD6) (RC3030): MQRCCF_MSG_SEQ_NUMBER_ERROR

Explanation

Message sequence number not valid.

The message sequence number parameter value was not valid.

Programmer response

Specify a valid message sequence number.

3031 (0BD7) (RC3031): MQRCCF_PING_DATA_COUNT_ERROR

Explanation

Data count not valid.

The Ping Channel *DataCount* value was not valid.

Programmer response

Specify a valid data count value.

3032 (0BD8) (RC3032): MQRCCF_PING_DATA_COMPARE_ERROR

Explanation

Ping Channel command failed.

The Ping Channel command failed with a data compare error. The data offset that failed is returned in the message (with parameter identifier MQIACF_ERROR_OFFSET).

Programmer response

Consult your systems administrator.

3033 (0BD9) (RC3033): MQRCCF_CFSL_PARM_ID_ERROR

Explanation

Parameter identifier is not valid.

The MQCFSL *Parameter* field value was not valid.

Programmer response

Specify a valid parameter identifier.

3034 (0BDA) (RC3034): MQRCCF_CHANNEL_TYPE_ERROR

Explanation

Channel type not valid.

The *ChannelType* specified was not valid, or did not match the type of an existing channel being copied, changed or replaced, or the command and the specified disposition cannot be used with that type of channel.

Programmer response

Specify a valid channel name, type, or disposition.

3035 (0BDB) (RC3035): MQRCCF_PARM_SEQUENCE_ERROR

Explanation

Parameter sequence not valid.

The sequence of parameters is not valid for this command.

Programmer response

Specify the positional parameters in a valid sequence for the command.

3036 (0BDC) (RC3036): MQRCCF_XMIT_PROTOCOL_TYPE_ERR

Explanation

Transmission protocol type not valid.

The *TransportType* value was not valid.

Programmer response

Specify a valid transmission protocol type.

3037 (0BDD) (RC3037): MQRCCF_BATCH_SIZE_ERROR

Explanation

Batch size not valid.

The batch size specified was not valid.

Programmer response

Specify a valid batch size value.

3038 (0BDE) (RC3038): MQRCCF_DISC_INT_ERROR

Explanation

Disconnection interval not valid.

The disconnection interval specified was not valid.

Programmer response

Specify a valid disconnection interval.

3039 (0BDF) (RC3039): MQRCCF_SHORT_RETRY_ERROR

Explanation

Short retry count not valid.

The *ShortRetryCount* value was not valid.

Programmer response

Specify a valid short retry count value.

3040 (0BE0) (RC3040): MQRCCF_SHORT_TIMER_ERROR

Explanation

Short timer value not valid.

The *ShortRetryInterval* value was not valid.

Programmer response

Specify a valid short timer value.

3041 (0BE1) (RC3041): MQRCCF_LONG_RETRY_ERROR

Explanation

Long retry count not valid.

The long retry count value specified was not valid.

Programmer response

Specify a valid long retry count value.

3042 (0BE2) (RC3042): MQRCCF_LONG_TIMER_ERROR

Explanation

Long timer not valid.

The long timer (long retry wait interval) value specified was not valid.

Programmer response

Specify a valid long timer value.

3043 (0BE3) (RC3043): MQRCCF_SEQ_NUMBER_WRAP_ERROR

Explanation

Sequence wrap number not valid.

The *SeqNumberWrap* value was not valid.

Programmer response

Specify a valid sequence wrap number.

3044 (0BE4) (RC3044): MQRCCF_MAX_MSG_LENGTH_ERROR

Explanation

Maximum message length not valid.

The maximum message length value specified was not valid.

Programmer response

Specify a valid maximum message length.

3045 (0BE5) (RC3045): MQRCCF_PUT_AUTH_ERROR

Explanation

Put authority value not valid.

The *PutAuthority* value was not valid.

Programmer response

Specify a valid authority value.

3046 (0BE6) (RC3046): MQRCCF_PURGE_VALUE_ERROR

Explanation

Purge value not valid.

The *Purge* value was not valid.

Programmer response

Specify a valid purge value.

3047 (0BE7) (RC3047): MQRCCF_CFIL_PARM_ID_ERROR

Explanation

Parameter identifier is not valid.

The MQCFIL or MQCFIL64 *Parameter* field value was not valid, or specifies a parameter that cannot be filtered, or that is also specified as a parameter to select a subset of objects.

Programmer response

Specify a valid parameter identifier.

3048 (0BE8) (RC3048): MQRCCF_MSG_TRUNCATED

Explanation

Message truncated.

The command server received a message that is larger than its maximum valid message size.

Programmer response

Check the message contents are correct.

3049 (0BE9) (RC3049): MQRCCF_CCSID_ERROR

Explanation

Coded character-set identifier error.

In a command message, one of the following occurred:

- The *CodedCharSetId* field in the message descriptor of the command does not match the coded character-set identifier of the queue manager at which the command is being processed, or
- The *CodedCharSetId* field in a string parameter structure within the message text of the command is not
  - MQCCSI_DEFAULT, or
  - the coded character-set identifier of the queue manager at which the command is being processed, as in the *CodedCharSetId* field in the message descriptor.

The error response message contains the correct value.

This reason can also occur if a ping cannot be performed because the coded character-set identifiers are not compatible. In this case the correct value is not returned.

Programmer response

Construct the command with the correct coded character-set identifier, and specify this in the message descriptor when sending the command. For ping, use a suitable coded character-set identifier.

3050 (0BEA) (RC3050): MQRCCF_ENCODING_ERROR

Explanation

Encoding error.

The *Encoding* field in the message descriptor of the command does not match that required for the platform at which the command is being processed.

Programmer response

Construct the command with the correct encoding, and specify this in the message descriptor when sending the command.

3052 (0BEC) (RC3052): MQRCCF_DATA_CONV_VALUE_ERROR

Explanation

Data conversion value not valid.

The value specified for *DataConversion* is not valid.

Programmer response

Specify a valid value.

3053 (0BED) (RC3053): MQRCCF_INDOUBT_VALUE_ERROR

Explanation

In-doubt value not valid.

The value specified for *InDoubt* is not valid.

Programmer response

Specify a valid value.

3054 (0BEE) (RC3054): MQRCCF_ESCAPE_TYPE_ERROR

Explanation

Escape type not valid.

The value specified for *EscapeType* is not valid.

Programmer response

Specify a valid value.

3062 (0BF6) (RC3062): MQRCCF_CHANNEL_TABLE_ERROR

Explanation

Channel table value not valid.

The *ChannelTable* specified was not valid, or was not appropriate for the channel type specified on an Inquire Channel or Inquire Channel Names command.

Programmer response

Specify a valid channel table value.

3063 (0BF7) (RC3063): MQRCCF_MCA_TYPE_ERROR

Explanation

Message channel agent type not valid.

The *MCAType* value specified was not valid.

Programmer response

Specify a valid value.

3064 (0BF8) (RC3064): MQRCCF_CHL_INST_TYPE_ERROR

Explanation

Channel instance type not valid.

The *ChannelInstanceType* specified was not valid.

Programmer response

Specify a valid channel instance type.

3065 (0BF9) (RC3065): MQRCCF_CHL_STATUS_NOT_FOUND

Explanation

Channel status not found.

For Inquire Channel Status, no channel status is available for the specified channel. This might indicate that the channel has not been used.

Programmer response

None, unless this is unexpected, in which case consult your systems administrator.

3066 (0BFA) (RC3066): MQRCCF_CFSL_DUPLICATE_PARM

Explanation

Duplicate parameter.

Two MQCFSL structures, or an MQCFST followed by an MQCFSL structure, with the same parameter identifier were present.

Programmer response

Check for and remove duplicate parameters.

3067 (0BFB) (RC3067): MQRCCF_CFSL_TOTAL_LENGTH_ERROR

Explanation

Total string length error.

The total length of the strings (not including trailing blanks) in a MQCFSL structure exceeds the maximum allowable for the parameter.

Programmer response

Check that the structure has been specified correctly, and if so reduce the number of strings.

3068 (0BFC) (RC3068): MQRCCF_CFSL_COUNT_ERROR

Explanation

Count of parameter values not valid.

The MQCFSL *Count* field value was not valid. The value was negative or greater than the maximum permitted for the parameter specified in the *Parameter* field.

Programmer response

Specify a valid count for the parameter.

3069 (0BFD) (RC3069): MQRCCF_CFSL_STRING_LENGTH_ERR

Explanation

String length not valid.

The MQCFSL *StringLength* field value was not valid. The value was negative or greater than the maximum permitted length of the parameter specified in the *Parameter* field.

Programmer response

Specify a valid string length for the parameter.

3070 (0BFE) (RC3070): MQRCCF_BROKER_DELETED

Explanation

Broker has been deleted.

When a broker is deleted using the *dltmqbrk* command, all broker queues created by the broker are deleted. Before this can be done the queues are emptied of all command messages; any that are found are placed on the dead-letter queue with this reason code.

Programmer response

Process the command messages that were placed on the dead-letter queue.

3071 (0BFF) (RC3071): MQRCCF_STREAM_ERROR

Explanation

Stream name is not valid.

The stream name parameter is not valid. Stream names must obey the same naming rules as for IBM MQ queues.

Programmer response

Retry the command with a valid stream name parameter.

3072 (0C00) (RC3072): MQRCCF_TOPIC_ERROR

Explanation

Topic name is invalid.

A command has been sent to the broker containing a topic name that is not valid. Note that wildcard topic names are not allowed for *Register Publisher* and *Publish* commands.

Programmer response

Retry the command with a valid topic name parameter. Up to 256 characters of the topic name in question are returned with the error response message. If the topic name contains a null character, this is assumed to terminate the string and is not considered to be part of it. A zero length topic name is not valid, as is one that contains an escape sequence that is not valid.

3073 (0C01) (RC3073): MQRCCF_NOT_REGISTERED

Explanation

Subscriber or publisher is not registered.

A *Deregister* command has been issued to remove registrations for a topic, or topics, for which the publisher or subscriber is not registered. If multiple topics were specified on the command, it fails with a completion code of MQCC_WARNING if the publisher or subscriber was registered for some, but not all, of the topics specified. This error code is also returned to a subscriber issuing a *Request Update* command for a topic for which he does not have a subscription.

Programmer response

Investigate why the publisher or subscriber is not registered. In the case of a subscriber, the subscriptions might have expired, or been removed automatically by the broker if the subscriber is no longer authorized.

3074 (0C02) (RC3074): MQRCCF_Q_MGR_NAME_ERROR

Explanation

An invalid or unknown queue manager name has been supplied.

A queue manager name has been supplied as part of a publisher or subscriber identity. This might have been supplied as an explicit parameter or in the *ReplyToQMgr* field in the message descriptor of the command. Either the queue manager name is not valid, or in the case of a subscriber identity, the subscriber's queue could not be resolved because the remote queue manager is not known to the broker queue manager.

Programmer response

Retry the command with a valid queue manager name. If appropriate, the broker includes a further error reason code within the error response message. If one is supplied, follow the guidance for that reason code in "Messages and reason codes" on page 244 to resolve the problem.

3075 (0C03) (RC3075): MQRCCF_INCORRECT_STREAM

Explanation

Stream name does not match the stream queue it was sent to.

A command has been sent to a stream queue that specified a different stream name parameter.

Programmer response

Retry the command either by sending it to the correct stream queue or by modifying the command so that the stream name parameter matches.

3076 (0C04) (RC3076): MQRCCF_Q_NAME_ERROR

Explanation

An invalid or unknown queue name has been supplied.

A queue name has been supplied as part of a publisher or subscriber identity. This might have been supplied as an explicit parameter or in the *ReplyToQ* field in the message descriptor of the command.

Either the queue name is not valid, or in the case of a subscriber identity, the broker has failed to open the queue.

Programmer response

Retry the command with a valid queue name. If appropriate, the broker includes a further error reason code within the error response message. If one is supplied, follow the guidance for that reason code in “Messages and reason codes” on page 244 to resolve the problem.

3077 (0C05) (RC3077): MQRCCF_NO_RETAINED_MSG

Explanation

No retained message exists for the topic specified.

A *Request Update* command has been issued to request the retained message associated with the specified topic. No retained message exists for that topic.

Programmer response

If the topic or topics in question should have retained messages, the publishers of these topics might not be publishing with the correct publication options to cause their publications to be retained.

3078 (0C06) (RC3078): MQRCCF_DUPLICATE_IDENTITY

Explanation

Publisher or subscriber identity already assigned to another user ID.

Each publisher and subscriber has a unique identity consisting of a queue manager name, a queue name, and optionally a correlation identifier. Associated with each identity is the user ID under which that publisher or subscriber first registered. A specific identity can be assigned only to one user ID at a time. While the identity is registered with the broker all commands wanting to use it must specify the correct user ID. When a publisher or a subscriber no longer has any registrations with the broker the identity can be used by another user ID.

Programmer response

Either retry the command using a different identity or remove all registrations associated with the identity so that it can be used by a different user ID. The user ID to which the identity is currently assigned is returned within the error response message. A *Deregister* command could be issued to remove these registrations. If the user ID in question cannot be used to execute such a command, you need to have the necessary authority to open the SYSTEM.BROKER.CONTROL.QUEUE using the MQOO_ALTERNATE_USER_AUTHORITY option.

3079 (0C07) (RC3079): MQRCCF_INCORRECT_Q

Explanation

Command sent to wrong broker queue.

The command is a valid broker command but the queue it has been sent to is incorrect. *Publish* and *Delete Publication* commands need to be sent to the stream queue, all other commands need to be sent to the SYSTEM.BROKER.CONTROL.QUEUE.

Programmer response

Retry the command by sending it to the correct queue.

3080 (0C08) (RC3080): MQRCCF_CORREL_ID_ERROR

Explanation

Correlation identifier used as part of an identity is all binary zeros.

Each publisher and subscriber is identified by a queue manager name, a queue name, and optionally a correlation identifier. The correlation identifier is typically used to allow multiple subscribers to share the same subscriber queue. In this instance a publisher or subscriber has indicated within the Registration or Publication options supplied on the command that their identity does include a correlation identifier, but a valid identifier has not been supplied. The <RegOpt>CorrelAsId</RegOpt> has been specified, but the correlation identifier of the message is nulls.

Programmer response

Change the program to retry the command ensuring that the correlation identifier supplied in the message descriptor of the command message is not all binary zeros.

3081 (0C09) (RC3081): MQRCCF_NOT_AUTHORIZED

Explanation

Subscriber has insufficient authority.

To receive publications a subscriber application needs both browse authority for the stream queue that it is subscribing to, and put authority for the queue that publications are to be sent to. Subscriptions are rejected if the subscriber does not have both authorities. In addition to having browse authority for the stream queue, a subscriber would also require *altusr* authority for the stream queue to subscribe to certain topics that the broker itself publishes information on. These topics start with the MQ/SA/ prefix.

Programmer response

Ensure that the subscriber has the necessary authorities and reissue the request. The problem might occur because the subscriber's user ID is not known to the broker. This can be identified if a further error reason code of MQRC_UNKNOWN_ENTITY is returned within the error response message.

3082 (0C0A) (RC3082): MQRCCF_UNKNOWN_STREAM

Explanation

Stream is not known by the broker or could not be created.

A command message has been put to the SYSTEM.BROKER.CONTROL.QUEUE for an unknown stream. This error code is also returned if dynamic stream creation is enabled and the broker failed to create a stream queue for the new stream using the SYSTEM.BROKER.MODEL.STREAM queue.

Programmer response

Retry the command for a stream that the broker supports. If the broker should support the stream, either define the stream queue manually, or correct the problem that prevented the broker from creating the stream queue itself.

3083 (0C0B) (RC3083): MQRCCF_REG_OPTIONS_ERROR

Explanation

Invalid registration options have been supplied.

The registration options (between <RegOpt> and </RegOpt>) provided on a command are not valid.

Programmer response

Retry the command with a valid combination of options.

3084 (0C0C) (RC3084): MQRCCF_PUB_OPTIONS_ERROR

Explanation

Invalid publication options have been supplied.

The publication options provided on a Publish command are not valid.

Programmer response

Retry the command with a valid combination of options.

3085 (0C0D) (RC3085): MQRCCF_UNKNOWN_BROKER

Explanation

Command received from an unknown broker.

Within a multi-broker network, related brokers pass subscriptions and publications between each other as a series of command messages. One such command message has been received from a broker that is not, or is no longer, related to the detecting broker.

Programmer response

This situation can occur if the broker network is not quiesced while topology changes are made to the network.

If you are removing a broker from the topology when the queue manager is inactive, your changes are propagated at queue manager restart.

If you are removing a broker from the topology when the queue manager is active, make sure the channels are also active, so that your changes are immediately propagated.

3086 (0C0E) (RC3086): MQRCCF_Q_MGR_CCSID_ERROR

Explanation

Queue manager coded character set identifier error.

The coded character set value for the queue manager was not valid.

Programmer response

Specify a valid value.

3087 (0C0F) (RC3087): MQRCCF_DEL_OPTIONS_ERROR

Explanation

Invalid delete options have been supplied.

The options provided with a *Delete Publication* command are not valid.

Programmer response

Retry the command with a valid combination of options.

3088 (0C10) (RC3088): MQRCCF_CLUSTER_NAME_CONFLICT

Explanation

ClusterName and **ClusterNamelist** attributes conflict.

The command was rejected because it would have resulted in the **ClusterName** attribute and the **ClusterNamelist** attribute both being nonblank. At least one of these attributes must be blank.

Programmer response

If the command specified one of these attributes only, you must also specify the other one, but with a value of blanks. If the command specified both attributes, ensure that one of them has a value of blanks.

3089 (0C11) (RC3089): MQRCCF_REPOS_NAME_CONFLICT

Explanation

RepositoryName and **RepositoryNamelist** attributes conflict.

Either:

- The command was rejected because it would have resulted in the **RepositoryName** and **RepositoryNamelist** attributes both being nonblank. At least one of these attributes must be blank.
- For a Reset Queue Manager Cluster command, the queue manager does not provide a full repository management service for the specified cluster. That is, the **RepositoryName** attribute of the queue manager is not the specified cluster name, or the namelist specified by the **RepositoryNamelist** attribute does not contain the cluster name.

Programmer response

Reissue the command with the correct values or on the correct queue manager.

3090 (0C12) (RC3090): MQRCCF_CLUSTER_Q_USAGE_ERROR

Explanation

Queue cannot be a cluster queue.

The command was rejected because it would have resulted in a cluster queue also being a transmission queue, which is not permitted, or because the queue in question cannot be a cluster queue.

Programmer response

Ensure that the command specifies either:

- The **Usage** parameter with a value of MQUS_NORMAL, or
- The **ClusterName** and **ClusterNamelist** parameters with values of blanks.
- A **QName** parameter with a value that is not one of these reserved queues:
  - SYSTEM.CHANNEL.INITQ
  - SYSTEM.CHANNEL.SYNCQ
  - SYSTEM.CLUSTER.COMMAND.QUEUE
  - SYSTEM.CLUSTER.REPOSITORY.QUEUE
  - SYSTEM.COMMAND.INPUT
  - SYSTEM.QSG.CHANNEL.SYNCQ
  - SYSTEM.QSG.TRANSMIT.QUEUE

3091 (0C13) (RC3091): MQRCCF_ACTION_VALUE_ERROR

Explanation

Action value not valid.

The value specified for *Action* is not valid. There is only one valid value.

Programmer response

Specify MQACT_FORCE_REMOVE as the value of the **Action** parameter.

3092 (0C14) (RC3092): MQRCCF_COMMS_LIBRARY_ERROR

Explanation

Library for requested communications protocol could not be loaded.

The library needed for the requested communications protocol could not be loaded.

Programmer response

Install the library for the required communications protocol, or specify a communications protocol that has already been installed.

3093 (0C15) (RC3093): MQRCCF_NETBIOS_NAME_ERROR

Explanation

NetBIOS listener name not defined.

The NetBIOS listener name is not defined.

Programmer response

Add a local name to the configuration file and retry the operation.

3094 (0C16) (RC3094): MQRCCF_BROKER_COMMAND_FAILED

Explanation

The broker command failed to complete.

A broker command was issued but it failed to complete.

Programmer response

Diagnose the problem using the provided information and issue a corrected command.

For more information, look at the IBM MQ error logs.

3095 (0C17) (RC3095): MQRCCF_CFST_CONFLICTING_PARM

Explanation

Conflicting parameters.

The command was rejected because the parameter identified in the error response was in conflict with another parameter in the command.

Programmer response

Consult the description of the parameter identified to ascertain the nature of the conflict, and the correct command.

3096 (0C18) (RC3096): MQRCCF_PATH_NOT_VALID

Explanation

Path not valid.

The path specified was not valid.

Programmer response

Specify a valid path.

3097 (0C19) (RC3097): MQRCCF_PARM_SYNTAX_ERROR

Explanation

The value specified for the parameter contains a syntax error.

The value might not conform to one of the following generic syntax rules:

- The value contains an asterisk (*) which is not the last character.
- The value contains a question mark (?) or colon (:).
- The keyword is WHERE and the value is a single asterisk.

Programmer response

Check the syntax for this parameter.

3098 (0C1A) (RC3098): MQRCCF_PWD_LENGTH_ERROR

Explanation

Password length error.

The password string length is rounded up by to the nearest eight bytes. This rounding causes the total length of the *SSLCryptoHardware* string to exceed its maximum.

Programmer response

Decrease the size of the password, or of earlier fields in the *SSLCryptoHardware* string.

3150 (0C4E) (RC3150): MQRCCF_FILTER_ERROR

Explanation

Filter not valid. This could be because either:

1. In an inquire command message, the specification of a filter is not valid.
2. In a publish/subscribe command message, the content-based filter expression supplied in the publish/subscribe command message contains invalid syntax, and cannot be used.

Programmer response

1. Correct the specification of the filter parameter structure in the inquire command message.
2. Correct the syntax of the filter expression in the publish/subscribe command message. The filter expression is the value of the *Filter* tag in the *psc* folder in the MQRFH2 structure. See the *WebSphere MQ Integrator V2 Programming Guide* for details of valid syntax.

3151 (0C4F) (RC3151): MQRCCF_WRONG_USER

Explanation

Wrong user.

A publish/subscribe command message cannot be executed on behalf of the requesting user because the subscription that it would update is already owned by a different user. A subscription can be updated or deregistered only by the user that originally registered the subscription.

Programmer response

Ensure that applications that need to issue commands against existing subscriptions are running under the user identifier that originally registered the subscription. Alternatively, use different subscriptions for different users.

3152 (0C50) (RC3152): MQRCCF_DUPLICATE_SUBSCRIPTION

Explanation

The subscription already exists.

A matching subscription already exists.

Programmer response

Either modify the new subscription properties to distinguish it from the existing subscription or deregister the existing subscription. Then reissue the command.

3153 (0C51) (RC3153): MQRCCF_SUB_NAME_ERROR

Explanation

The subscription name parameter is in error.

Either the subscription name is of an invalid format or a matching subscription already exists with no subscription name.

Programmer response

Either correct the subscription name or remove it from the command and reissue the command.

3154 (0C52) (RC3154): MQRCCF_SUB_IDENTITY_ERROR

Explanation

The subscription identity parameter is in error.

Either the supplied value exceeds the maximum length allowed or the subscription identity is not currently a member of the subscription's identity set and a Join registration option was not specified.

Programmer response

Either correct the identity value or specify a Join registration option to add this identity to the identity set for this subscription.

3155 (0C53) (RC3155): MQRCCF_SUBSCRIPTION_IN_USE

Explanation

The subscription is in use.

An attempt to modify or deregister a subscription was attempted by a member of the identity set when they were not the only member of this set.

Programmer response

Reissue the command when you are the only member of the identity set. To avoid the identity set check and force the modification or deregistration remove the subscription identity from the command message and reissue the command.

3156 (0C54) (RC3156): MQRCCF_SUBSCRIPTION_LOCKED

Explanation

The subscription is locked.

The subscription is currently exclusively locked by another identity.

Programmer response

Wait for this identity to release the exclusive lock.

3157 (0C55) (RC3157): MQRCCF_ALREADY_JOINED

Explanation

The identity already has an entry for this subscription.

A Join registration option was specified but the subscriber identity was already a member of the subscription's identity set.

Programmer response

None. The command completed, this reason code is a warning.

3160 (0C58) (RC3160): MQRCCF_OBJECT_IN_USE

Explanation

Object in use by another command.

A modification of an object was attempted while the object was being modified by another command.

Programmer response

Retry the command.

3161 (0C59) (RC3161): MQRCCF_UNKNOWN_FILE_NAME

Explanation

File not defined to CICS.

A file name parameter identifies a file that is not defined to CICS.

Programmer response

Provide a valid file name or create a CSD definition for the required file.

3162 (0C5A) (RC3162): MQRCCF_FILE_NOT_AVAILABLE

Explanation

File not available to CICS.

A file name parameter identifies a file that is defined to CICS, but is not available.

Programmer response

Check that the CSD definition for the file is correct and enabled.

3163 (0C5B) (RC3163): MQRCCF_DISC_RETRY_ERROR

Explanation

Disconnection retry count not valid.

The *DiscRetryCount* value was not valid.

Programmer response

Specify a valid count.

3164 (0C5C) (RC3164): MQRCCF_ALLOC_RETRY_ERROR

Explanation

Allocation retry count not valid.

The *AllocRetryCount* value was not valid.

Programmer response

Specify a valid count.

3165 (0C5D) (RC3165): MQRCCF_ALLOC_SLOW_TIMER_ERROR

Explanation

Allocation slow retry timer value not valid.

The *AllocRetrySlowTimer* value was not valid.

Programmer response

Specify a valid timer value.

3166 (0C5E) (RC3166): MQRCCF_ALLOC_FAST_TIMER_ERROR

Explanation

Allocation fast retry timer value not valid.

The *AllocRetryFastTimer* value was not valid.

Programmer response

Specify a valid value.

3167 (0C5F) (RC3167): MQRCCF_PORT_NUMBER_ERROR

Explanation

Port number value not valid.

The *PortNumber* value was not valid.

Programmer response

Specify a valid port number value.

3168 (0C60) (RC3168): MQRCCF_CHL_SYSTEM_NOT_ACTIVE

Explanation

Channel system is not active.

An attempt was made to start a channel while the channel system was inactive.

Programmer response

Activate the channel system before starting a channel.

3169 (0C61) (RC3169): MQRCCF_ENTITY_NAME_MISSING

Explanation

Entity name required but missing.

A parameter specifying entity names must be supplied.

Programmer response

Specify the required parameter.

3170 (0C62) (RC3170): MQRCCF_PROFILE_NAME_ERROR

Explanation

Profile name not valid.

A profile name is not valid. Profile names might include wildcard characters or might be given explicitly. If you give an explicit profile name, then the object identified by the profile name must exist. This error might also occur if you specify more than one double asterisk in a profile name.

Programmer response

Specify a valid name.

3171 (0C63) (RC3171): MQRCCF_AUTH_VALUE_ERROR

Explanation

Authorization value not valid.

A value for the **AuthorizationList** or **AuthorityRemove** or **AuthorityAdd** parameter was not valid.

Programmer response

Specify a valid value.

3172 (0C64) (RC3172): MQRCCF_AUTH_VALUE_MISSING

Explanation

Authorization value required but missing.

A parameter specifying authorization values must be supplied.

Programmer response

Specify the required parameter.

3173 (0C65) (RC3173): MQRCCF_OBJECT_TYPE_MISSING

Explanation

Object type value required but missing.

A parameter specifying the object type must be supplied.

Programmer response

Specify the required parameter.

3174 (0C66) (RC3174): MQRCCF_CONNECTION_ID_ERROR

Explanation

Error in connection ID parameter.

The *ConnectionId* specified was not valid.

Programmer response

Specify a valid connection ID.

3175 (0C67) (RC3175): MQRCCF_LOG_TYPE_ERROR

Explanation

Log type not valid.

The log type value specified was not valid.

Programmer response

Specify a valid log type value.

3176 (0C68) (RC3176): MQRCCF_PROGRAM_NOT_AVAILABLE

Explanation

Program not available.

A request to start or stop a service failed because the request to start the program failed. This could be because the program could not be found at the specified location, or that insufficient system resources are available currently to start it.

Programmer response

Check that the correct name is specified in the definition of the service, and that the program is in the appropriate libraries, before retrying the request.

3177 (0C69) (RC3177): MQRCCF_PROGRAM_AUTH_FAILED

Explanation

Program not available.

A request to start or stop a service failed because the user does not have sufficient access authority to start the program at the specified location.

Programmer response

Correct the progam name and location, and the user's authority, before retrying the request.

3200 (0C80) (RC3200): MQRCCF_NONE_FOUND

Explanation

No items found matching request criteria.

An Inquire command found no items that matched the specified name and satisfied any other criteria requested.

3201 (0C81) (RC3201): MQRCCF_SECURITY_SWITCH_OFF

Explanation

Security refresh or reverification not processed, security switch set OFF.

Either

- a Reverify Security command was issued, but the subsystem security switch is off, so there are no internal control tables to flag for reverification; or
- a Refresh Security command was issued, but the security switch for the requested class or the subsystem security switch is off.

The switch in question might be returned in the message (with parameter identifier MQIACF_SECURITY_SWITCH).

3202 (0C82) (RC3202): MQRCCF_SECURITY_REFRESH_FAILED

Explanation

Security refresh did not take place.

A SAF RACROUTE REQUEST=STAT call to your external security manager (ESM) returned a non-zero return code. In consequence, the requested security refresh could not be done. The security item affected might be returned in the message (with parameter identifier MQIACF_SECURITY_ITEM).

Possible causes of this problem are:

- The class is not installed
- The class is not active
- The external security manager (ESM) is not active
- The RACF z/OS router table is incorrect

Programmer response

For information about resolving the problem, see the explanations of messages CSQH003I and CSQH004I.

3203 (0C83) (RC3203): MQRCCF_PARM_CONFLICT

Explanation

Incompatible parameters or parameter values.

The parameters or parameter values for a command are incompatible. One of the following occurred:

- A parameter was not specified that is required by another parameter or parameter value.
- A parameter or parameter value was specified that is not allowed with some other parameter or parameter value.
- The values for two specified parameters were not both blank or non-blank.
- The values for two specified parameters were incompatible.
- The specified value is inconsistent with the configuration.

The parameters in question might be returned in the message (with parameter identifiers MQIACF_PARAMETER_ID).

Programmer response

Reissue the command with correct parameters and values.

3204 (0C84) (RC3204): MQRCCF_COMMAND_INHIBITED

Explanation

Commands not allowed at present time.

The queue manager cannot accept commands at the present time, because it is restarting or terminating, or because the command server is not running.

3205 (0C85) (RC3205): MQRCCF_OBJECT_BEING_DELETED

Explanation

Object is being deleted.

The object specified on a command is in the process of being deleted, so the command is ignored.

3207 (0C87) (RC3207): MQRCCF_STORAGE_CLASS_IN_USE

Explanation

Storage class is active or queue is in use.

The command for a local queue involved a change to the *StorageClass* value, but there are messages on the queue, or other threads have the queue open.

Programmer response

Remove the messages from the queue, or wait until any other threads have closed the queue.

3208 (0C88) (RC3208): MQRCCF_OBJECT_NAME_RESTRICTED

Explanation

Incompatible object name and type.

The command used a reserved object name with an incorrect object type or subtype. The object is only allowed to be of a predetermined type, as listed in the explanation of message CSQM108I.

3209 (0C89) (RC3209): MQRCCF_OBJECT_LIMIT_EXCEEDED

Explanation

Local queue limit exceeded.

The command failed because no more local queues could be defined. There is an implementation limit of 524 287 for the total number of local queues that can exist. For shared queues, there is a limit of 512 queues in a single coupling facility structure.

Programmer response

Delete any existing queues that are no longer required.

3210 (0C8A) (RC3210): MQRCCF_OBJECT_OPEN_FORCE

Explanation

Object is in use, but could be changed specifying *Force* as MQFC_YES.

The object specified is in use. This could be because it is open through the API, or for certain parameter changes, because there are messages currently on the queue. The requested changes can be made by specifying *Force* as MQFC_YES on a Change command.

Programmer response

Wait until the object is not in use. Alternatively specify *Force* as MQFC_YES for a change command.

3211 (0C8B) (RC3211): MQRCCF_DISPOSITION_CONFLICT

Explanation

Parameters are incompatible with disposition.

The parameters or parameter values for a command are incompatible with the disposition of an object. One of the following occurred:

- A value specified for the object name or other parameter is not allowed for a local queue with a disposition that is shared or a model queue used to create a dynamic queue that is shared.
- A value specified for a parameter is not allowed for an object with such disposition.
- A value specified for a parameter must be non-blank for an object with such disposition.
- The **CommandScope** and **QSGDisposition** or **ChannelDisposition** parameter values are incompatible.
- The action requested for a channel cannot be performed because it has the wrong disposition.

The parameter and disposition in question may be returned in the message (with parameter identifiers MQIACF_PARAMETER_ID and MQIA_QSG_DISP).

Programmer response

Reissue the command with correct parameters and values.

3212 (0C8C) (RC3212): MQRCCF_Q_MGR_NOT_IN_QSG

Explanation

Queue manager is not in a queue sharing group.

The command or its parameters are not allowed when the queue manager is not in a queue sharing group. The parameter in question might be returned in the message (with parameter identifier MQIACF_PARAMETER_ID).

Programmer response

Reissue the command correctly.

3213 (0C8D) (RC3213): MQRCCF_ATTR_VALUE_FIXED

Explanation

Parameter value cannot be changed.

The value for a parameter cannot be changed. The parameter in question might be returned in the message (with parameter identifier MQIACF_PARAMETER_ID).

Programmer response

To change the parameter, the object must be deleted and then created again with the new value.

3215 (0C8F) (RC3215): MQRCCF_NAMELIST_ERROR

Explanation

Namelist is empty or wrong type.

A namelist used to specify a list of clusters has no names in it or does not have type MQNT_CLUSTER or MQNT_NONE.

Programmer response

Reissue the command specifying a namelist that is not empty and has a suitable type.

3217 (0C91) (RC3217): MQRCCF_NO_CHANNEL_INITIATOR

Explanation

Channel initiator not active.

The command requires the channel initiator to be started.

3218 (0C93) (RC3218): MQRCCF_CHANNEL_INITIATOR_ERROR

Explanation

Channel initiator cannot be started, or no suitable channel initiator is available.

This might occur because of the following reasons:

- The channel initiator cannot be started because:
  - It is already active.
  - There are insufficient system resources.
  - The queue manager was shutting down.
- The shared channel cannot be started because there was no suitable channel initiator available for any active queue manager in the queue sharing group. This could be because:
  - No channel initiators are running.
  - The channel initiators that are running are too busy to allow any channel, or a channel of the particular type, to be started.

3222 (0C96) (RC3222): MQRCCF_COMMAND_LEVEL_CONFLICT

Explanation

Incompatible queue manager command levels.

Changing the **CFLevel** parameter of a CF structure, or deleting a CF structure, requires that all queue managers in the queue sharing group have a command level of at least 530. Some of the queue managers have a level less than 530.

3223 (0C97) (RC3223): MQRCCF_Q_ATTR_CONFLICT

Explanation

Queue attributes are incompatible.

The queues involved in a Move Queue command have different values for one or more of these attributes: *DefinitionType*, *HardenGetBackout*, *Usage*. Messages cannot be moved safely if these attributes differ.

3224 (0C98) (RC3224): MQRCCF_EVENTS_DISABLED

Explanation

Events not enabled.

The command required performance or configuration events to be enabled.

Programmer response

Use the Change Queue manager command to enable the events if required.

3225 (0C99) (RC3225): MQRCCF_COMMAND_SCOPE_ERROR

Explanation

Queue sharing group error.

While processing a command that used the **CommandScope** parameter, an error occurred while trying to send data to the coupling facility.

Programmer response

Notify your system programmer.

3226 (0C9A) (RC3226): MQRCCF_COMMAND_REPLY_ERROR

Explanation

Error saving command reply information.

While processing a command that used the **CommandScope** parameter, or a command for the channel initiator, an error occurred while trying to save information about the command.

Programmer response

The most likely cause is insufficient storage. If the problem persists, you may need to restart the queue manager after making more storage available.

3227 (0C9B) (RC3227): MQRCCF_FUNCTION_RESTRICTED

Explanation

Restricted command or parameter value used.

The command, or the value specified for one of its parameters, is not allowed because the installation and customization options chosen do not allow all functions to be used. The parameter in question might be returned in the message (with parameter identifier MQIACF_PARAMETER_ID).

3228 (0C9C) (RC3228): MQRCCF_PARM_MISSING

Explanation

Required parameter not specified.

The command did not specify a parameter or parameter value that was required. It might be for one of the following reasons:

- A parameter that is always required.
- A parameter that is one of a set of two or more alternative required parameters.
- A parameter that is required because some other parameter was specified.
- A parameter that is a list of values which has too few values.

The parameter in question might be returned in the message (with parameter identifier MQIACF_PARAMETER_ID).

Programmer response

Reissue the command with correct parameters and values.

3229 (0C9D) (RC3229): MQRCCF_PARM_VALUE_ERROR

Explanation

Parameter value invalid.

The value specified for a parameter was not acceptable. It might be for one of the following reasons:

- Outside the acceptable numeric range for the parameter.

- Not one of a list of acceptable values for the parameter.
- Using characters that are invalid for the parameter.
- Completely blank, when such is not allowed for the parameter.
- A filter value that is invalid for the parameter being filtered.

The parameter in question might be returned in the message (with parameter identifier MQIACF_PARAMETER_ID).

Programmer response

Reissue the command with correct parameters and values.

3230 (0C9E) (RC3230): MQRCCF_COMMAND_LENGTH_ERROR

Explanation

Command exceeds allowable length.

The command is so large that its internal form has exceeded the maximum length allowed. The size of the internal form of the command is affected by both the length, and the complexity of the command.

z/OS This return code is also returned on z/OS, if the internal form of the response is larger than supported when using CMDSCOPE.

3231 (0C9F) (RC3231): MQRCCF_COMMAND_ORIGIN_ERROR

Explanation

Command issued incorrectly.

The command cannot be issued using command server. This is an internal error.

Programmer response

Notify your system programmer.

3232 (0CA0) (RC3232): MQRCCF_LISTENER_CONFLICT

Explanation

Address conflict for listener.

A listener was already active for a port and IP address combination that conflicted with the *Port* and *IPAddress* values specified by a Start Channel Listener or Stop Channel Listener command. The *Port* and *IPAddress* value combination specified must match a combination for which the listener is active. It cannot be a superset or a subset of that combination.

Programmer response

Reissue the command with correct values, if required.

3233 (0CA1) (RC3233): MQRCCF_LISTENER_STARTED

Explanation

Listener is started.

An attempt was made to start a listener, but it is already active for the requested *TransportType*, *InboundDisposition*, *Port*, and *IPAddress* values. The requested parameter values might be returned in the message, if applicable (with parameter identifiers MQIACH_XMIT_PROTOCOL_TYPE, MQIACH_INBOUND_DISP, MQIACH_PORT_NUMBER, MQCACH_IP_ADDRESS).

3234 (0CA2) (RC3234): MQRCCF_LISTENER_STOPPED

Explanation

Listener is stopped.

An attempt was made to stop a listener, but it is not active or already stopping for the requested *TransportType*, *InboundDisposition*, *Port*, and *IPAddress* values. The requested parameter values might be returned in the message, if applicable (with parameter identifiers MQIACH_XMIT_PROTOCOL_TYPE, MQIACH_INBOUND_DISP, MQIACH_PORT_NUMBER, MQCACH_IP_ADDRESS).

3235 (0CA3) (RC3235): MQRCCF_CHANNEL_ERROR

Explanation

Channel command failed.

A channel command failed because of an error in the channel definition, or at the remote end of the channel, or in the communications system. An error identifier value *nnn* may be returned in the message (with parameter identifier MQIACF_ERROR_ID).

Programmer response

For information about the error, see the explanation of the corresponding error message. Error *nnn* generally corresponds to message CSQX *nnn*, although there are some exceptions. z/OS For more information, see "Distributed queuing message codes" on page 1147.

3236 (0CA4) (RC3236): MQRCCF_CF_STRUC_ERROR

Explanation

CF structure error.

A command could not be processed because of a coupling facility or CF structure error. It might be:

- A Backup CF Structure or Recover CF Structure command when the status of the CF structure is unsuitable. In this case, the CF structure status might be returned in the message together with the CF structure name (with parameter identifiers MQIACF_CF_STRUC_STATUS and MQCA_CF_STRUC_NAME).
- A command could not access an object because of an error in the coupling facility information, or because a CF structure has failed. In this case, the name of the object involved might be returned in the message (with parameter identifier MQCA_Q_NAME, for example).
- A command involving a shared channel could not access the channel status or synchronization key information.

Programmer response

In the case of a Backup CF Structure or Recover CF Structure command, take action appropriate to the CF structure status reported.

In other cases, check for error messages on the console log that might relate to the problem. Check whether the coupling facility structure has failed and check that Db2 is available.

3237 (0CA5) (RC3237): MQRCCF_UNKNOWN_USER_ID

Explanation

User identifier not found.

A user identifier specified in a Reverify Security command was not valid because there was no entry found for it in the internal control table. This could be because the identifier was entered incorrectly in the command, or because it was not in the table (for example, because it had timed-out). The user identifier in question might be returned in the message (with parameter identifier MQCACF_USER_IDENTIFIER).

3238 (0CA6) (RC3238): MQRCCF_UNEXPECTED_ERROR

Explanation

Unexpected or severe error.

An unexpected or severe error or other failure occurred. A code associated with the error might be returned in the message (with parameter identifier MQIACF_ERROR_ID).

Programmer response

Notify your system programmer.

3239 (0CA7) (RC3239): MQRCCF_NO_XCF_PARTNER

Explanation

MQ is not connected to the XCF partner.

The command involving the IMS bridge cannot be processed because MQ is not connected to the XCF partner. The group and member names of the XCF partner in question might be returned in the message (with parameter identifiers MQCA_XCF_GROUP_NAME and MQCA_XCF_MEMBER_NAME).

3240 (0CA8) (RC3240): MQRCCF_CFGR_PARM_ID_ERROR

Explanation

Parameter identifier is not valid.

The MQCFGR *Parameter* field value was not valid.

Programmer response

Specify a valid parameter identifier.

3241 (0CA9) (RC3241): MQRCCF_CFIF_LENGTH_ERROR

Explanation

Structure length not valid.

The MQCFIF *StrucLength* field value was not valid.

Programmer response

Specify a valid structure length.

3242 (0CAA) (RC3242): MQRCCF_CFIF_OPERATOR_ERROR

Explanation

Parameter count not valid.

The MQCFIF *Operator* field value was not valid.

Programmer response

Specify a valid operator value.

3243 (0CAB) (RC3243): MQRCCF_CFIF_PARM_ID_ERROR

Explanation

Parameter identifier is not valid.

The MQCFIF *Parameter* field value was not valid, or specifies a parameter that cannot be filtered, or that is also specified as a parameter to select a subset of objects.

Programmer response

Specify a valid parameter identifier.

3244 (0CAC) (RC3244): MQRCCF_CFSF_FILTER_VAL_LEN_ERR

Explanation

Filter value length not valid.

The MQCFSF *FilterValueLength* field value was not valid.

Programmer response

Specify a valid length.

3245 (0CAD) (RC3245): MQRCCF_CFSF_LENGTH_ERROR

Explanation

Structure length not valid.

The MQCFSF *StrucLength* field value was not valid.

Programmer response

Specify a valid structure length.

3246 (0CAE) (RC3246): MQRCCF_CFSF_OPERATOR_ERROR

Explanation

Parameter count not valid.

The MQCFSF *Operator* field value was not valid.

Programmer response

Specify a valid operator value.

3247 (0CAF) (RC3247): MQRCCF_CFSF_PARM_ID_ERROR

Explanation

Parameter identifier is not valid.

The MQCFSF *Parameter* field value was not valid.

Programmer response

Specify a valid parameter identifier.

3248 (0CB0) (RC3248): MQRCCF_TOO_MANY_FILTERS

Explanation

Too many filters.

The command contained more than the maximum permitted number of filter structures.

Programmer response

Specify the command correctly.

3249 (0CB1) (RC3249): MQRCCF_LISTENER_RUNNING

Explanation

Listener is running.

An attempt was made to perform an operation on a listener, but it is currently active.

Programmer response

Stop the listener if required.

3250 (0CB2) (RC3250): MQRCCF_LSTR_STATUS_NOT_FOUND

Explanation

Listener status not found.

For Inquire Listener Status, no listener status is available for the specified listener. This might indicate that the listener has not been used.

Programmer response

None, unless this is unexpected, in which case consult your systems administrator.

3251 (0CB3) (RC3251): MQRCCF_SERVICE_RUNNING

Explanation

Service is running.

An attempt was made to perform an operation on a service, but it is currently active.

Programmer response

Stop the service if required.

3252 (0CB4) (RC3252): MQRCCF_SERV_STATUS_NOT_FOUND

Explanation

Service status not found.

For Inquire Service Status, no service status is available for the specified service. This might indicate that the service has not been used.

Programmer response

None, unless this is unexpected, in which case consult your systems administrator.

3253 (0CB5) (RC3253): MQRCCF_SERVICE_STOPPED

Explanation

Service is stopped.

An attempt was made to stop a service, but it is not active or already stopping.

3254 (0CB6) (RC3254): MQRCCF_CFBS_DUPLICATE_PARM

Explanation

Duplicate parameter.

Two MQCFBS structures with the same parameter identifier were present.

Programmer response

Check for and remove duplicate parameters.

3255 (0CB7) (RC3255): MQRCCF_CFBS_LENGTH_ERROR

Explanation

Structure length not valid.

The MQCFBS *StrucLength* field value was not valid.

Programmer response

Specify a valid structure length.

3256 (0CB8) (RC3256): MQRCCF_CFBS_PARM_ID_ERROR

Explanation

Parameter identifier is not valid.

The MQCFBS *Parameter* field value was not valid.

Programmer response

Specify a valid parameter identifier.

3257 (0CB9) (RC3257): MQRCCF_CFBS_STRING_LENGTH_ERR

Explanation

String length not valid.

The MQCFBS *StringLength* field value was not valid. The value was negative or greater than the maximum permitted length of the parameter specified in the *Parameter* field.

Programmer response

Specify a valid string length for the parameter.

3258 (0CBA) (RC3258): MQRCCF_CFGR_LENGTH_ERROR

Explanation

Structure length not valid.

The MQCFGR *StrucLength* field value was not valid.

Programmer response

Specify a valid structure length.

3259 (0CBB) (RC3259): MQRCCF_CFGR_PARM_COUNT_ERROR

Explanation

Parameter count not valid.

The MQCFGR *ParameterCount* field value was not valid. The value was negative or greater than the maximum permitted for the parameter identifier specified in the *Parameter* field.

Programmer response

Specify a valid count for the parameter.

3260 (0CBC) (RC3260): MQRCCF_CONN_NOT_STOPPED

Explanation

Connection not stopped.

The Stop Connection command could not be executed, so the connection was not stopped.

3261 (0CBD) (RC3261): MQRCCF_SERVICE_REQUEST_PENDING

Explanation

A Suspend or Resume Queue Manager command was issued, or a Refresh Security command, but such a command is currently in progress.

Programmer response

Wait until the current request completes, then reissue the command if necessary.

3262 (0CBE) (RC3262): MQRCCF_NO_START_CMD

Explanation

No start command.

The service cannot be started because no start command is specified in the service definition.

Programmer response

Correct the definition of the service.

3263 (0CBF) (RC3263): MQRCCF_NO_STOP_CMD

Explanation

No stop command.

The service cannot be stopped because no stop command is specified in the service definition.

Programmer response

Correct the definition of the service.

3264 (0CC0) (RC3264): MQRCCF_CFBF_LENGTH_ERROR

Explanation

Structure length not valid.

The MQCFBF *StrucLength* field value was not valid.

Programmer response

Specify a valid structure length.

3265 (0CC1) (RC3265): MQRCCF_CFBF_PARM_ID_ERROR

Explanation

Parameter identifier is not valid.

The MQCFBF *Parameter* field value was not valid.

Programmer response

Specify a valid parameter identifier.

3266 (0CC2) (RC3266): MQRCCF_CFBF_FILTER_VAL_LEN_ERR

Explanation

Filter value length not valid.

The MQCFBF *FilterValueLength* field value was not valid.

Programmer response

Specify a valid length.

3267 (0CC3) (RC3267): MQRCCF_CFBF_OPERATOR_ERROR

Explanation

Parameter count not valid.

The MQCFBF *Operator* field value was not valid.

Programmer response

Specify a valid operator value.

3268 (0CC4) (RC3268): MQRCCF_LISTENER_STILL_ACTIVE

Explanation

Listener still active.

An attempt was made to stop a listener, but it failed and the listener is still active. For example, the listener might still have active channels.

Programmer response

Wait for the active connections to the listener to complete before trying the request again.

3269 (0CC5) (RC3269): MQRCCF_DEF_XMIT_Q_CLUS_ERROR

Explanation

The specified queue is not allowed to be used as the default transmission queue because it is reserved for use exclusively by clustering.

Programmer response

Change the value of the Default Transmission Queue, and try the command again.

3300 (0CE4) (RC3300): MQRCCF_TOPICSTR_ALREADY_EXISTS

Explanation

The topic string specified already exists in another topic object.

Programmer response

Verify that the topic string used is correct.

3301 (0CE5) (RC3301): MQRCCF_SHARING_CONVS_ERROR

Explanation

An invalid value has been given for SharingConversations parameter in the Channel definition

Programmer response

Correct the value used in the PCF SharingConversations (MQCFIN) parameter; see Change, Copy, and Create Channel for more information.

3302 (0CE6) (RC3302): MQRCCF_SHARING_CONVS_TYPE

Explanation

SharingConversations parameter is not allowed for this channel type.

Programmer response

See Change, Copy, and Create Channel to ensure that the channel type is compatible with the SharingConversations parameter.

3303 (0CE7) (RC3303): MQRCCF_SECURITY_CASE_CONFLICT

Explanation

A Refresh Security PCF command was issued, but the case currently in use differs from the system setting and if refreshed would result in the set of classes using different case settings.

Programmer response

Check that the class used is set up correctly and that the system setting is correct. If a change in case setting is required, issue the REFRESH SECURITY(*) command to change all classes.

3305 (0CE9) (RC3305): MQRCCF_TOPIC_TYPE_ERROR

Explanation

An Inquire or Delete Topic PCF command was issued with an invalid TopicType parameter.

Programmer response

Correct the TopicType parameter and reissue the command. For more details on the TopicType, see Change, Copy, and Create Topic.

3306 (0CEA) (RC3306): MQRCCF_MAX_INSTANCES_ERROR

Explanation

An invalid value was given for the maximum number of simultaneous instances of a server-connection channel (MaxInstances) for the channel definition.

Programmer response

See Change, Copy, and Create Channel for more information and correct the PCF application.

3307 (0CEB) (RC3307): MQRCCF_MAX_INSTS_PER_CLNT_ERR

Explanation

An invalid value was given for the MaxInstancesPerClient property.

Programmer response

See Change, Copy, and Create Channel for the range of values and correct the application.

3308 (0CEC) (RC3308): MQRCCF_TOPIC_STRING_NOT_FOUND

Explanation

When processing an Inquire Topic Status command, the topic string specified did not match any topic nodes in the topic tree.

Programmer response

Verify the topic string is correct.

3309 (0CED) (RC3309): MQRCCF_SUBSCRIPTION_POINT_ERR

Explanation

The Subscription point was not valid. Valid subscription points are the topic strings of the topic objects listed in the SYSTEM.QPUBSUB.SUBPOINT.NAMELIST.

Programmer response

Use a subscription point that matches the topic string of a topic object listed in the SYSTEM.QPUBSUB.SUBPOINT.NAMELIST (or remove the subscription point parameter and this uses the default subscription point)

3311 (0CEF) (RC2432): MQRCCF_SUB_ALREADY_EXISTS

Explanation

When processing a Copy or Create Subscription command, the target *Subscription* identifier exists.

Programmer response

If you are trying to copy an existing subscription, ensure that the *ToSubscriptionName* parameter contains a unique value. If you are trying to create a Subscription ensure that the combination of the *SubName* parameter, and *TopicObject* parameter or *TopicString* parameter are unique.

3314 (0CF2) (RC3314): MQRCCF_DURABILITY_NOT_ALLOWED

Explanation

An MQSUB call using the MQSO_DURABLE option failed. This can be for one of the following reasons:

- The topic subscribed to is defined as DURSUB(NO).

- The queue named SYSTEM.DURABLE.SUBSCRIBER.QUEUE is not available.
- The topic subscribed to is defined as both MCAST(ONLY) and DURSUB(YES) (or DURSUB(ASPARENT) and the parent is DURSUB(YES)).

Completion code

MQCC_FAILED

Programmer response

Durable subscriptions are stored on the SYSTEM.DURABLE.SUBSCRIBER.QUEUE. Ensure that this queue is available for use. Possible reasons for failure include the queue being full, the queue being put inhibited, the queue not existing, or (on z/OS) the page set the queue is defined to use doesn't exist.

If the topic subscribed to is defined as DURSUB(NO) either alter the administrative topic node to use DURSUB(YES) or use the MQSO_NON_DURABLE option instead.

If the topic subscribed to is defined as MCAST(ONLY) when using IBM MQ Multicast messaging, alter the topic to use DURSUB(NO).

3317 (0CF5) (RC3317): MQRCCF_INVALID_DESTINATION

Explanation

The Subscription or Topic object used in a Change, Copy, Create or Delete PCF command is invalid.

Programmer response

Investigate and correct the required parameters for the specific command you are using. For more details, see Change, Copy, and Create Subscription.

3318 (0CF6) (RC3318): MQRCCF_PUBSUB_INHIBITED

Explanation

MQSUB, MQOPEN, MQPUT and MQPUT1 calls are currently inhibited for all publish/subscribe topics, either by means of the queue manager attribute PSMODE or because processing of publish/subscribe state at queue manager start-up has failed, or has not yet completed.

Completion code

MQCC_FAILED

Programmer response

If this queue manager does not intentionally inhibit publish/subscribe, investigate any error messages that describe the failure at queue manager start-up, or wait until start-up processing completes. You can use the DISPLAY PUBSUB command to check the status of the publish/subscribe engine to ensure it is ready for use, and additionally on z/OS you will receive an information message CSQM076I.

3326 (0CFE) (RC3326): MQRCCF_CHLAUTH_TYPE_ERROR

Explanation

Channel authentication record type not valid.

The **type** parameter specified on the **set** command was not valid.

Programmer response
Specify a valid type.

3327 (0CFF) (RC3327): MQRCCF_CHLAUTH_ACTION_ERROR

Explanation
Channel authentication record action not valid.

The **action** parameter specified on the **set** command was not valid.

Programmer response
Specify a valid action.

3335 (0D07) (RC3335): MQRCCF_CHLAUTH_USRSRC_ERROR

Explanation
Channel authentication record user source not valid.

The **user source** parameter specified on the **set** command was not valid.

Programmer response
Specify a valid user source.

3336 (0D08) (RC3336): MQRCCF_WRONG_CHLAUTH_TYPE

Explanation
Parameter not allowed for this channel authentication record type.

The parameter is not allowed for the type of channel authentication record being set. Refer to the description of the parameter in error to determine the types of record for which this parameter is valid.

Programmer response
Remove the parameter.

3337 (0D09) (RC3337): MQRCCF_CHLAUTH_ALREADY_EXISTS

Explanation
Channel authentication record already exists

An attempt was made to add a channel authentication record, but it already exists.

Programmer response
Specify action as MQACT_REPLACE.

3338 (0D0A) (RC3338): MQRCCF_CHLAUTH_NOT_FOUND

Explanation
Channel authentication record not found.

The specified channel authentication record does not exist.

Programmer response
Specify a channel authentication record that exists.

3339 (0D0B) (RC3339): MQRCCF_WRONG_CHLAUTH_ACTION

Explanation
Parameter not allowed for this action on a channel authentication record.

The parameter is not allowed for the action being applied to a channel authentication record. Refer to the description of the parameter in error to determine the actions for which this parameter is valid.

Programmer response
Remove the parameter.

3340 (0D0C) (RC3340): MQRCCF_WRONG_CHLAUTH_USERSRC

Explanation
Parameter not allowed for this channel authentication record user source value.

The parameter is not allowed for a channel authentication record with the value that the **user source** field contains. Refer to the description of the parameter in error to determine the values of user source for which this parameter is valid.

Programmer response
Remove the parameter.

3341 (0D0D) (RC3341): MQRCCF_CHLAUTH_WARN_ERROR

Explanation
Channel authentication record **warn** value not valid.

The **warn** parameter specified on the **set** command was not valid.

Programmer response
Specify a valid value for **warn**.

3342 (0D0E) (RC3342): MQRCCF_WRONG_CHLAUTH_MATCH

Explanation
Parameter not allowed for this channel authentication record **match** value.

The parameter is not allowed for an **inquire channel authentication record** command with the value that the **match** field contains. Refer to the description of the parameter in error to find the values of **match** for which this parameter is valid.

Programmer response
Remove the parameter.

3343 (0D0F) (RC3343): MQRCCF_IPADDR_RANGE_CONFLICT

Explanation
A channel authentication record contained an IP address with a range that overlapped an existing range. A range must be a superset or subset of any existing ranges for the same channel profile name, or completely separate.

Programmer response
Specify a range that is a superset or subset of an existing range, or is completely separate to all existing ranges.

3344 (0D10) (RC3344): MQRCCF_CHLAUTH_MAX_EXCEEDED

Explanation
A channel authentication record was set taking the total number of entries for that type on a single channel profile over the maximum number allowed.

Programmer response
Remove some channel authentication records to make room.

3345 (0D11) (RC3345): MQRCCF_IPADDR_ERROR

Explanation
A channel authentication record contained an invalid IP address, or invalid wildcard pattern to match against IP addresses.

Programmer response
Specify a valid IP address or pattern.
Related reference
Generic IP addresses

3346 (0D12) (RC3346): MQRCCF_IPADDR_RANGE_ERROR

Explanation
A channel authentication record contained an IP address with a range that was invalid, for example, the smaller number is greater than or equal to the larger number of the range.

Programmer response
Specify a valid range in the IP address.

3347 (0D13) (RC3347): MQRCCF_PROFILE_NAME_MISSING

Explanation

Profile name missing.

A profile name was required for the command but none was specified.

Programmer response

Specify a valid profile name.

3348 (0D14) (RC3348): MQRCCF_CHLAUTH_CLNTUSER_ERROR

Explanation
Channel authentication record **client user** value not valid.

The **client user** value contains a wildcard character, which is not allowed.

Programmer response
Specify a valid value for the client user field.

3349 (0D15) (RC3349): MQRCCF_CHLAUTH_NAME_ERROR

Explanation
Channel authentication record channel name not valid.

When a channel authentication record specifies an IP address to block, the **channel name** value must be a single asterisk (*).

Programmer response
Enter a single asterisk in the channel name.

3350 (0D16) (RC3350): MQRCCF_CHLAUTH_RUNCHECK_ERROR

Runcheck command is using generic values.

Explanation

An Inquire Channel Authentication Record command using MQMATCH_RUNCHECK was issued, but one or more of the input fields on the command were provided with generic values, which is not allowed.

Programmer response

Enter non-generic values for channel name, address, one of the client user ID or remote queue manager and TLS Peer Name if used.

3353 (0D19) (RC3353): MQRCCF_SUITE_B_ERROR

Invalid values have been specified.

Explanation

An invalid combination of values has been specified for the **MQIA_SUITE_B_STRENGTH** parameter.

Programmer response

Review the combination entered and retry with appropriate values.

3363 (0D23) (RC3363): MQRCCF_CLUS_XMIT_Q_USAGE_ERROR

Explanation

If the local queue attribute **CLCHNAME** is set, the attribute **USAGE** must be set to XMITQ.

z/OS On z/OS, if the local queue attribute **CLCHNAME** is set, the attribute **INDXTYPE** must be set to **CORRELID**, and the transmission queue must not be a shared queue.

The **CLCHNAME** attribute is a generic cluster-sender channel name. It identifies the cluster-sender channel that transfers messages in a transmission queue to another queue manager.

Programmer response

Modify the application to set the **CLCHNAME** to blanks, or not set the **CLCHNAME** attribute at all, on queues other than transmission queues.

z/OS On z/OS, ensure that the transmission queue is indexed by correlation ID and the queue is not a shared queue.

3364 (0D24) (RC3364): MQRCCF_CERT_VAL_POLICY_ERROR

Explanation

An invalid certificate validation policy value was specified for the **MQIA_CERT_VAL_POLICY** attribute. The specified value is unknown or is not supported on the current platform.

Programmer response

Review the value specified and try again with an appropriate certificate validation policy.

3366 (0D26) (RC3366): MQRCCF_REVDNS_DISABLED

Explanation

A runcheck command completed successfully returning the records to be used. However, some Channel Authentication Records exist containing host names, and host name reverse lookup is currently disabled, so these records will not have been matched against. This reason code is returned as an MQCC_WARNING.

Programmer response

If reverse lookup is correctly disabled, even though some Channel Authentication Records exist containing host names, this warning can be ignored.

If channel authentication records containing host names should be being matched against, and therefore reverse lookup of host name should not currently be disabled, issue a Change Queue Manager command to re-enable it.

If reverse lookup for host names is correctly disabled and there should not be any Channel Authentication Records containing host names, issue a Set Channel Authentication Record to remove them.

3370 (0D2A) (RC3370): MQRCCF_CHLAUTH_CHKCLI_ERROR

Explanation

Channel authentication record check client not valid.

The check client parameter specified on the set command was not valid.

Programmer response

Specify a valid user source.

3377 (0D31) (RC3377): MQRCCF_TOPIC_RESTRICTED

Explanation

This error can occur when creating or modifying a topic object. One or more attributes of the topic object are not supported on an IBM MQ administrative topic.

Programmer response

Modify the configuration to adhere to the documented restrictions.

z/OS 3389 (0D3D) (RC3389): MQRCCF_INCOMPATIBLE_QM_IN_QSG

Explanation

An attempt was made to set attribute *attribute* on a shared object when one or more queue managers in the queue sharing group are running a version of IBM MQ for z/OS that does not support *attribute*. The object remains unchanged.

Programmer response

Adjust your command so *attribute* is not set, or upgrade all the queue managers in the queue sharing group to a version of IBM MQ for z/OS that does support *attribute*.

CAPEXPRY is supported from IBM MQ for z/OS 9.4.0.

V 10.0.0 **OTELTRAC** and **OTELPCTL** are supported from IBM MQ for z/OS 10.0.0. This includes support on topic objects.

V 10.0.0 z/OS 3390 (0D3E) (RC3390): MQRCCF_ATTR_VALUE_ERROR_QSG_QM

Explanation

One or more queue managers in the queue sharing group is earlier than IBM MQ 10.0.0, and a **ConnectionName** with a length greater than 48 characters was specified on a channel with a **QSGDisposition** of either MQQSGD_COPY or MQQSGD_GROUP.

Programmer response

Either use a shorter **ConnectionName** or ensure that all queue managers in the queue sharing group are at IBM MQ 10.0.0 or later.

V 10.0.0 Multi 3391 (0D3F) RC3391): MQRCCF_AUTHORIZED

Explanation

This occurs within an authority event message when authorization has been successfully granted to the queue manager or subsequent IBM MQ object.

Programmer response

None.

4001 (0FA1) (RC4001): MQRCCF_OBJECT_ALREADY_EXISTS

Explanation

Object already exists.

An attempt was made to create an object, but the object already existed and the **Replace** parameter was not specified as MQRP_YES.

Programmer response

Specify *Replace* as MQRP_YES, or use a different name for the object to be created.

4002 (0FA2) (RC4002): MQRCCF_OBJECT_WRONG_TYPE

Explanation

Object has wrong type or disposition.

An object already exists with the same name but a different subtype or disposition from that specified by the command.

Programmer response

Ensure that the specified object is the same subtype and disposition.

4003 (0FA3) (RC4003): MQRCCF_LIKE_OBJECT_WRONG_TYPE

Explanation

New and existing objects have different subtype.

An attempt was made to create an object based on the definition of an existing object, but the new and existing objects had different subtypes.

Programmer response

Ensure that the new object has the same subtype as the one on which it is based.

4004 (0FA4) (RC4004): MQRCCF_OBJECT_OPEN

Explanation

Object is open.

An attempt was made to operate on an object that was in use.

Programmer response

Wait until the object is not in use, and then retry the operation. Alternatively specify *Force* as MQFC_YES for a change command.

4005 (0FA5) (RC4005): MQRCCF_ATTR_VALUE_ERROR

Explanation

Attribute value not valid or repeated.

One or more of the attribute values specified are not valid or are repeated. The error response message contains the failing attribute selectors (with parameter identifier MQIACF_PARAMETER_ID).

Programmer response

Specify the attribute values correctly.

4006 (0FA6) (RC4006): MQRCCF_UNKNOWN_Q_MGR

Explanation

Queue manager not known.

The queue manager specified was not known.

Programmer response

Specify the name of the queue manager to which the command is sent, or blank.

4007 (0FA7) (RC4007): MQRCCF_Q_WRONG_TYPE

Explanation

Action not valid for the queue of specified type.

An attempt was made to perform an action on a queue of the wrong type.

Programmer response

Specify a queue of the correct type.

4008 (0FA8) (RC4008): MQRCCF_OBJECT_NAME_ERROR

Explanation

Name not valid.

An object or other name name was specified using characters that were not valid.

Programmer response

Specify only valid characters for the name.

4009 (0FA9) (RC4009): MQRCCF_ALLOCATE_FAILED

Explanation

Allocation failed.

An attempt to allocate a conversation to a remote system failed. The error might be due to an entry in the channel definition that is not valid, or it might be that the listening program at the remote system is not running.

Programmer response

Ensure that the channel definition is correct, and start the listening program if necessary. If the error persists, consult your systems administrator.

4010 (0FAA) (RC4010): MQRCCF_HOST_NOT_AVAILABLE

Explanation

Remote system not available.

An attempt to allocate a conversation to a remote system was unsuccessful. The error might be transitory, and the allocate might succeed later. This reason can occur if the listening program at the remote system is not running.

Programmer response

Ensure that the listening program is running, and retry the operation.

4011 (0FAB) (RC4011): MQRCCF_CONFIGURATION_ERROR

Explanation

Configuration error.

There was a configuration error in the channel definition or communication subsystem, and allocation of a conversation was not possible. This might be caused by one of the following:

- For LU 6.2, either the *ModeName* or the *TpName* is incorrect. The *ModeName* must match that on the remote system, and the *TpName* must be specified. (On IBM i, these are held in the communications Side Object.)
- For LU 6.2, the session might not be established.
- For TCP, the *ConnectionName* in the channel definition cannot be resolved to a network address. This might be because the name has not been correctly specified, or because the name server is not available.
- The requested communications protocol might not be supported on the platform.

Programmer response

Identify the error and take appropriate action.

4012 (0FAC) (RC4012): MQRCCF_CONNECTION_REFUSED

Explanation

Connection refused.

The attempt to establish a connection to a remote system was rejected. The remote system might not be configured to allow a connection from this system.

- For LU 6.2 either the user ID or the password supplied to the remote system is incorrect.
- For TCP the remote system might not recognize the local system as valid, or the TCP listener program might not be started.

Programmer response

Correct the error or restart the listener program.

4013 (0FAD) (RC4013): MQRCCF_ENTRY_ERROR

Explanation

Connection name not valid.

The connection name in the channel definition could not be resolved into a network address. Either the name server does not contain the entry, or the name server was not available.

Programmer response

Ensure that the connection name is correctly specified and that the name server is available.

4014 (0FAE) (RC4014): MQRCCF_SEND_FAILED

Explanation

Send failed.

An error occurred while sending data to a remote system. This might be caused by a communications failure.

Programmer response

Consult your systems administrator.

4015 (0FAF) (RC4015): MQRCCF_RECEIVED_DATA_ERROR

Explanation

Received data error.

An error occurred while receiving data from a remote system. This might be caused by a communications failure.

Programmer response

Consult your systems administrator.

4016 (0FB0) (RC4016): MQRCCF_RECEIVE_FAILED

Explanation

Receive failed.

The receive operation failed.

Programmer response

Correct the error and retry the operation.

4017 (0FB1) (RC4017): MQRCCF_CONNECTION_CLOSED

Explanation

Connection closed.

An error occurred while receiving data from a remote system. The connection to the remote system has unexpectedly terminated.

Programmer response

Contact your systems administrator.

4018 (0FB2) (RC4018): MQRCCF_NO_STORAGE

Explanation

Not enough storage available.

Insufficient storage is available.

Programmer response

Consult your systems administrator.

4019 (0FB3) (RC4019): MQRCCF_NO_COMMS_MANAGER

Explanation

Communications manager not available.

The communications subsystem is not available.

Programmer response

Ensure that the communications subsystem has been started.

4020 (0FB4) (RC4020): MQRCCF_LISTENER_NOT_STARTED

Explanation

Listener not started.

The listener program could not be started. Either the communications subsystem has not been started, or the number of current channels using the communications subsystem is the maximum allowed, or there are too many jobs waiting in the queue.

Programmer response

Ensure the communications subsystem is started or retry the operation later. Increase the number of current channels allowed, if appropriate.

4024 (0FB8) (RC4024): MQRCCF_BIND_FAILED

Explanation

Bind failed.

The bind to a remote system during session negotiation has failed.

Programmer response

Consult your systems administrator.

4025 (0FB9) (RC4025): MQRCCF_CHANNEL_INDOUBT

Explanation

Channel in-doubt.

The requested operation cannot complete because the channel is in doubt.

Programmer response

Examine the status of the channel, and either restart a channel to resolve the in-doubt state, or resolve the channel.

4026 (0FBA) (RC4026): MQRCCF_MQCONN_FAILED

Explanation

MQCONN call failed.

Programmer response

Check whether the queue manager is active.

4027 (0FBB) (RC4027): MQRCCF_MQOPEN_FAILED

Explanation

MQOPEN call failed.

Programmer response

Check whether the queue manager is active, and the queues involved are correctly set up.

4028 (0FBC) (RC4028): MQRCCF_MQGET_FAILED

Explanation

MQGET call failed.

Programmer response

Check whether the queue manager is active, and the queues involved are correctly set up, and enabled for MQGET.

4029 (0FBD) (RC4029): MQRCCF_MQPUT_FAILED

Explanation

MQPUT call failed.

Programmer response

Check whether the queue manager is active, and the queues involved are correctly set up, and not inhibited for puts.

4030 (0FBE) (RC4030): MQRCCF_PING_ERROR

Explanation

Ping error.

A ping operation can only be issued for a sender or server channel. If the local channel is a receiver channel, you must issue the ping from a remote queue manager.

Programmer response

Reissue the ping request for a different channel of the correct type, or for a receiver channel from a different queue manager.

4031 (0FBF) (RC4031): MQRCCF_CHANNEL_IN_USE

Explanation

Channel in use.

An attempt was made to perform an operation on a channel, but the channel is currently active.

Programmer response

Stop the channel or wait for it to terminate.

4032 (0FC0) (RC4032): MQRCCF_CHANNEL_NOT_FOUND

Explanation

Channel not found.

The channel specified does not exist.

Programmer response

Specify the name of a channel which exists.

4033 (0FC1) (RC4033): MQRCCF_UNKNOWN_REMOTE_CHANNEL

Explanation

Remote channel not known.

There is no definition of the referenced channel at the remote system.

Programmer response

Ensure that the local channel is correctly defined. If it is, add an appropriate channel definition at the remote system.

4034 (0FC2) (RC4034): MQRCCF_REMOTE_QM_UNAVAILABLE

Explanation

Remote queue manager not available.

The channel cannot be started because the remote queue manager is not available.

Programmer response

Start the remote queue manager.

4035 (0FC3) (RC4035): MQRCCF_REMOTE_QM_TERMINATING

Explanation

Remote queue manager terminating.

The channel is ending because the remote queue manager is terminating.

Programmer response

Restart the remote queue manager.

4036 (0FC4) (RC4036): MQRCCF_MQINQ_FAILED

Explanation

MQINQ call failed.

Programmer response

Check whether the queue manager is active.

4037 (0FC5) (RC4037): MQRCCF_NOT_XMIT_Q

Explanation

Queue is not a transmission queue.

The queue specified in the channel definition is not a transmission queue, or is in use.

Programmer response

Ensure that the queue is specified correctly in the channel definition, and that it is correctly defined to the queue manager.

4038 (0FC6) (RC4038): MQRCCF_CHANNEL_DISABLED

Explanation

Channel disabled.

An attempt was made to use a channel, but the channel was disabled (that is, stopped).

Programmer response

Start the channel.

4039 (0FC7) (RC4039): MQRCCF_USER_EXIT_NOT_AVAILABLE

Explanation

User exit not available.

The channel was terminated because the user exit specified does not exist.

Programmer response

Ensure that the user exit is correctly specified and the program is available.

4040 (0FC8) (RC4040): MQRCCF_COMMIT_FAILED

Explanation

Commit failed.

An error was received when an attempt was made to commit a unit of work.

Programmer response

Consult your systems administrator.

4041 (0FC9) (RC4041): MQRCCF_WRONG_CHANNEL_TYPE

Explanation

Parameter not allowed for this channel type.

The parameter is not allowed for the type of channel being created, copied, or changed. Refer to the description of the parameter in error to determine the types of channel for which the parameter is valid

Programmer response

Remove the parameter.

4042 (0FCA) (RC4042): MQRCCF_CHANNEL_ALREADY_EXISTS

Explanation

Channel already exists.

An attempt was made to create a channel but the channel already existed and *Replace* was not specified as MQRP_YES.

Programmer response

Specify *Replace* as MQRP_YES or use a different name for the channel to be created.

4043 (0FCB) (RC4043): MQRCCF_DATA_TOO_LARGE

Explanation

Data too large.

The data to be sent exceeds the maximum that can be supported for the command.

Programmer response

Reduce the size of the data.

4044 (0FCC) (RC4044): MQRCCF_CHANNEL_NAME_ERROR

Explanation

Channel name error.

The **ChannelName** parameter contained characters that are not allowed for channel names.

Programmer response

Specify a valid name.

4045 (0FCD) (RC4045): MQRCCF_XMIT_Q_NAME_ERROR

Explanation

Transmission queue name error.

The **XmitQName** parameter contains characters that are not allowed for queue names. This reason code also occurs if the parameter is not present when a sender or server channel is being created, and no default value is available.

Programmer response

Specify a valid name, or add the parameter.

4047 (0FCF) (RC4047): MQRCCF_MCA_NAME_ERROR

Explanation

Message channel agent name error.

The *MCAName* value contained characters that are not allowed for program names on the platform in question.

Programmer response

Specify a valid name.

4048 (0FD0) (RC4048): MQRCCF_SEND_EXIT_NAME_ERROR

Explanation

Channel send exit name error.

The *SendExit* value contained characters that are not allowed for program names on the platform in question.

Programmer response

Specify a valid name.

4049 (0FD1) (RC4049): MQRCCF_SEC_EXIT_NAME_ERROR

Explanation

Channel security exit name error.

The *SecurityExit* value contained characters that are not allowed for program names on the platform in question.

Programmer response

Specify a valid name.

4050 (0FD2) (RC4050): MQRCCF_MSG_EXIT_NAME_ERROR

Explanation

Channel message exit name error.

The *MsgExit* value contained characters that are not allowed for program names on the platform in question.

Programmer response

Specify a valid name.

4051 (0FD3) (RC4051): MQRCCF_RCV_EXIT_NAME_ERROR

Explanation

Channel receive exit name error.

The *ReceiveExit* value contained characters that are not allowed for program names on the platform in question.

Programmer response

Specify a valid name.

4052 (0FD4) (RC4052): MQRCCF_XMIT_Q_NAME_WRONG_TYPE

Explanation

Transmission queue name not allowed for this channel type.

The **XmitQName** parameter is only allowed for sender or server channel types.

Programmer response

Remove the parameter.

4053 (0FD5) (RC4053): MQRCCF_MCA_NAME_WRONG_TYPE

Explanation

Message channel agent name not allowed for this channel type.

The **MCAName** parameter is only allowed for sender, server or requester channel types.

Programmer response

Remove the parameter.

4054 (0FD6) (RC4054): MQRCCF_DISC_INT_WRONG_TYPE

Explanation

Disconnection interval not allowed for this channel type.

The **DiscInterval** parameter is only allowed for sender or server channel types.

Programmer response

Remove the parameter.

4055 (0FD7) (RC4055): MQRCCF_SHORT_RETRY_WRONG_TYPE

Explanation

Short retry parameter not allowed for this channel type.

The **ShortRetryCount** parameter is only allowed for sender or server channel types.

Programmer response

Remove the parameter.

4056 (0FD8) (RC4056): MQRCCF_SHORT_TIMER_WRONG_TYPE

Explanation

Short timer parameter not allowed for this channel type.

The **ShortRetryInterval** parameter is only allowed for sender or server channel types.

Programmer response

Remove the parameter.

4057 (0FD9) (RC4057): MQRCCF_LONG_RETRY_WRONG_TYPE

Explanation

Long retry parameter not allowed for this channel type.

The **LongRetryCount** parameter is only allowed for sender or server channel types.

Programmer response

Remove the parameter.

4058 (0FDA) (RC4058): MQRCCF_LONG_TIMER_WRONG_TYPE

Explanation

Long timer parameter not allowed for this channel type.

The **LongRetryInterval** parameter is only allowed for sender or server channel types.

Programmer response

Remove the parameter.

4059 (0FDB) (RC4059): MQRCCF_PUT_AUTH_WRONG_TYPE

Explanation

Put authority parameter not allowed for this channel type.

The **PutAuthority** parameter is only allowed for receiver or requester channel types.

Programmer response

Remove the parameter.

4061 (0FDD) (RC4061): MQRCCF_MISSING_CONN_NAME

Explanation

Connection name parameter required but missing.

The **ConnectionName** parameter is required for sender or requester channel types, but is not present.

Programmer response

Add the parameter.

4062 (0FDE) (RC4062): MQRCCF_CONN_NAME_ERROR

Explanation

Error in connection name parameter.

The **ConnectionName** parameter contains one or more blanks at the start of the name.

Programmer response

Specify a valid connection name.

4063 (0FDF) (RC4063): MQRCCF_MQSET_FAILED

Explanation

MQSET call failed.

Programmer response

Check whether the queue manager is active.

4064 (0FE0) (RC4064): MQRCCF_CHANNEL_NOT_ACTIVE

Explanation

Channel not active.

An attempt was made to stop a channel, but the channel was already stopped.

Programmer response

No action is required.

4065 (0FE1) (RC4065): MQRCCF_TERMINATED_BY_SEC_EXIT

Explanation

Channel terminated by security exit.

A channel security exit terminated the channel.

Programmer response

Check that the channel is attempting to connect to the correct queue manager, and if so that the security exit is specified correctly, and is working correctly, at both ends.

4067 (0FE3) (RC4067): MQRCCF_DYNAMIC_Q_SCOPE_ERROR

Explanation

Dynamic queue scope error.

The **Scope** attribute of the queue is to be MQSCO_CELL, but this is not allowed for a dynamic queue.

Programmer response

Predefine the queue if it is to have cell scope.

4068 (0FE4) (RC4068): MQRCCF_CELL_DIR_NOT_AVAILABLE

Explanation

Cell directory is not available.

The **Scope** attribute of the queue is to be MQSCO_CELL, but no name service supporting a cell directory has been configured.

Programmer response

Configure the queue manager with a suitable name service.

4069 (0FE5) (RC4069): MQRCCF_MR_COUNT_ERROR

Explanation

Message retry count not valid.

The *MsgRetryCount* value was not valid.

Programmer response

Specify a value in the range 0-999 999 999.

4070 (0FE6) (RC4070): MQRCCF_MR_COUNT_WRONG_TYPE

Explanation

Message-retry count parameter not allowed for this channel type.

The **MsgRetryCount** parameter is allowed only for receiver and requester channels.

Programmer response

Remove the parameter.

4071 (0FE7) (RC4071): MQRCCF_MR_EXIT_NAME_ERROR

Explanation

Channel message-retry exit name error.

The *MsgRetryExit* value contained characters that are not allowed for program names on the platform in question.

Programmer response

Specify a valid name.

4072 (0FE8) (RC4072): MQRCCF_MR_EXIT_NAME_WRONG_TYPE

Explanation

Message-retry exit parameter not allowed for this channel type.

The **MsgRetryExit** parameter is allowed only for receiver and requester channels.

Programmer response

Remove the parameter.

4073 (0FE9) (RC4073): MQRCCF_MR_INTERVAL_ERROR

Explanation

Message retry interval not valid.

The *MsgRetryInterval* value was not valid.

Programmer response

Specify a value in the range 0-999 999 999.

4074 (0FEA) (RC4074): MQRCCF_MR_INTERVAL_WRONG_TYPE

Explanation

Message-retry interval parameter not allowed for this channel type.

The **MsgRetryInterval** parameter is allowed only for receiver and requester channels.

Programmer response

Remove the parameter.

4075 (0FEB) (RC4075): MQRCCF_NPM_SPEED_ERROR

Explanation

Nonpersistent message speed not valid.

The *NonPersistentMsgSpeed* value was not valid.

Programmer response

Specify MQNPMS_NORMAL or MQNPMS_FAST.

4076 (0FEC) (RC4076): MQRCCF_NPM_SPEED_WRONG_TYPE

Explanation

Nonpersistent message speed parameter not allowed for this channel type.

The **NonPersistentMsgSpeed** parameter is allowed only for sender, receiver, server, requester, cluster sender, and cluster receiver channels.

Programmer response

Remove the parameter.

4077 (0FED) (RC4077): MQRCCF_HB_INTERVAL_ERROR

Explanation

Heartbeat interval not valid.

The *HeartbeatInterval* value was not valid.

Programmer response

Specify a value in the range 0-999 999.

4078 (0FEE) (RC4078): MQRCCF_HB_INTERVAL_WRONG_TYPE

Explanation

Heartbeat interval parameter not allowed for this channel type.

The **HeartbeatInterval** parameter is allowed only for receiver and requester channels.

Programmer response

Remove the parameter.

4079 (0FEF) (RC4079): MQRCCF_CHAD_ERROR

Explanation

Channel automatic definition error.

The *ChannelAutoDef* value was not valid.

Programmer response

Specify MQCHAD_ENABLED or MQCHAD_DISABLED.

4080 (0FF0) (RC4080): MQRCCF_CHAD_WRONG_TYPE

Explanation

Channel automatic definition parameter not allowed for this channel type.

The **ChannelAutoDef** parameter is allowed only for receiver and server-connection channels.

Programmer response

Remove the parameter.

4081 (0FF1) (RC4081): MQRCCF_CHAD_EVENT_ERROR

Explanation

Channel automatic definition event error.

The *ChannelAutoDefEvent* value was not valid.

Programmer response

Specify MQEVR_ENABLED or MQEVR_DISABLED.

4082 (0FF2) (RC4082): MQRCCF_CHAD_EVENT_WRONG_TYPE

Explanation

Channel automatic definition event parameter not allowed for this channel type.

The **ChannelAutoDefEvent** parameter is allowed only for receiver and server-connection channels.

Programmer response

Remove the parameter.

4083 (0FF3) (RC4083): MQRCCF_CHAD_EXIT_ERROR

Explanation

Channel automatic definition exit name error.

The *ChannelAutoDefExit* value contained characters that are not allowed for program names on the platform in question.

Programmer response

Specify a valid name.

4084 (0FF4) (RC4084): MQRCCF_CHAD_EXIT_WRONG_TYPE

Explanation

Channel automatic definition exit parameter not allowed for this channel type.

The **ChannelAutoDefExit** parameter is allowed only for receiver and server-connection channels.

Programmer response

Remove the parameter.

4085 (0FF5) (RC4085): MQRCCF_SUPPRESSED_BY_EXIT

Explanation

Action suppressed by exit program.

An attempt was made to define a channel automatically, but this was inhibited by the channel automatic definition exit. The **AuxErrorDataInt1** parameter contains the feedback code from the exit indicating why it inhibited the channel definition.

Programmer response

Examine the value of the **AuxErrorDataInt1** parameter, and take any action that is appropriate.

4086 (0FF6) (RC4086): MQRCCF_BATCH_INT_ERROR

Explanation

Batch interval not valid.

The batch interval specified was not valid.

Programmer response

Specify a valid batch interval value.

4087 (0FF7) (RC4087): MQRCCF_BATCH_INT_WRONG_TYPE

Explanation

Batch interval parameter not allowed for this channel type.

The **BatchInterval** parameter is allowed only for sender and server channels.

Programmer response

Remove the parameter.

4088 (0FF8) (RC4088): MQRCCF_NET_PRIORITY_ERROR

Explanation

Network priority value is not valid.

Programmer response

Specify a valid value.

4089 (0FF9) (RC4089): MQRCCF_NET_PRIORITY_WRONG_TYPE

Explanation

Network priority parameter not allowed for this channel type.

The **NetworkPriority** parameter is allowed for sender and server channels only.

Programmer response

Remove the parameter.

4090 (0FFA) (RC4090): MQRCCF_CHANNEL_CLOSED

Explanation

Channel closed.

The channel was closed prematurely. This can occur because a user stopped the channel while it was running, or a channel exit decided to close the channel.

Programmer response

Determine the reason that the channel was closed prematurely. Restart the channel if required.

4092 (0FFC) (RC4092): MQRCCF_SSL_CIPHER_SPEC_ERROR

Explanation

TLS cipher specification not valid.

The *SSLCipherSpec* specified is not valid.

Programmer response

Specify a valid cipher specification.

4093 (0FFD) (RC4093): MQRCCF_SSL_PEER_NAME_ERROR

Explanation

TLS peer name not valid.

The *SSLPeerName* specified is not valid.

Programmer response

Specify a valid peer name.

4094 (0FFE) (RC4094): MQRCCF_SSL_CLIENT_AUTH_ERROR

Explanation

TLS client authentication not valid.

The *SSLClientAuth* specified is not valid.

Programmer response

Specify a valid client authentication.

4095 (0FFF) (RC4095): MQRCCF_RETAINED_NOT_SUPPORTED

Explanation

Retained messages used on restricted stream.

An attempt has been made to use retained messages on a publish/subscribe stream defined to be restricted to JMS usage. JMS does not support the concept of retained messages and the request is rejected.

Programmer response

Either modify the application not to use retained messages, or modify the broker *JmsStreamPrefix* configuration parameter so that this stream is not treated as a JMS stream.

Linux AIX Token authentication error codes

Use the codes and explanations to understand and correct errors that can be caused when using authentication tokens.

These tables document error codes, in decimal form, that are displayed in messages AMQ5783E, AMQ5784E, and AMQ5786E.

Table 26. Error codes for authentication token format or validation errors

Error code (decimal)	Explanation	Response
101	A required claim or header parameter is missing from the authentication token. The name of the missing claim or header parameter is included in the message explanation.	Ensure that the authentication token contains all the required header parameters and claims. For more information about the requirements for authentication tokens that are supplied to IBM MQ, see Working with authentication tokens.
102	A claim or header parameter in the authentication token has an incorrect data type. For example, a string was specified as the value of claim that must be specified as an integer. The name of the claim or header parameter with the incorrect data type is included in the message explanation.	Ensure that the value of the specified header parameter or claim has the data type required by IBM MQ. For more information about the requirements for authentication tokens that are supplied to IBM MQ, see Working with authentication tokens.

Table 26. Error codes for authentication token format or validation errors (continued)

Error code (decimal)	Explanation	Response
103	The value of the authentication token's type header parameter is incorrect. The value of the token's "typ" (type) header parameter must be "JWT".	Obtain an authentication token with the correct value for the type header parameter.
104	The format of the authentication token does not conform to the JSON Web Signature (JWS) standard.	Obtain an authentication token that conforms to the JSON Web Signature (JWS) standard from your token issuer. That is, a JSON Web Token (JWT) that is secured with a digital signature or Message Authentication Code (MAC).
105	The header or payload of the authentication token is not valid JSON.	Obtain a correctly formatted JWT from your token issuer.
106	Validation of the authentication token's signature failed.	Ensure that the queue manager's token authentication key repository contains either the authentication token issuer's public key certificate or the symmetric key that is used to sign the authentication token. The **CertLabel** attribute in the **AuthToken** stanza of the `qm.ini` file must reference the label of the certificate or symmetric key. Issue the **REFRESH SECURITY TYPE(CONNAUTH)** MQSC command to bring changes to the queue manager's token authentication configuration into effect.
107	The queue manager is configured to adopt the user ID in the authentication token as the context for the application. The **UserClaim** attribute in the **AuthToken** stanza of the `qm.ini` file specifies the name of the claim that contains the user ID to be adopted. However, the claim is not present in the authentication token that is supplied by the application.	Obtain an authentication token that contains a claim with the name that the **UserClaim** attribute in the **AuthToken** stanza of the `qm.ini` file specifies. The value of the claim must be set to the user ID to be adopted as the context for the application.
108	The queue manager is configured to adopt the user ID in the authentication token as the context for the application. The **UserClaim** attribute in the **AuthToken** stanza of the `qm.ini` file specifies the name of the claim that contains the user ID to be adopted. However, the value of this claim in the authentication token does not meet the requirements for user IDs in authentication tokens.	Ensure that the value of the token user claim meets the restrictions for user IDs in authentication tokens. For more information, see User IDs in authentication tokens.

Table 26. Error codes for authentication token format or validation errors (continued)

Error code (decimal)	Explanation	Response
109	The algorithm used to sign the token is not supported. The value of the token's "alg" header parameter, which specifies the algorithm that is used to sign the token, is included in the message explanation.	Obtain a token from your token issuer that is signed with an algorithm that IBM MQ supports. For more information about the signing algorithms that are supported, see Working with authentication tokens.
110	No appropriate secret was found to validate an authentication token presented to the queue manager. The key ID claim name and value found in the token are included in the error message.	If the error returned to the client is MQRC_SECURITY_ERROR, check connectivity between your queue manager and authentication server. If the error returned to the client is MQRC_NOT_AUTHORIZED, confirm that this key is configured on the authentication server and included in the JKWS document it provides to configure token based authentication.
111	No corresponding issuer information found for the issuer set in the incoming JWT token.	Ensure that a JWKS stanza for this issuer name has been configured in the qm.ini file for this queue manager. If JWKS is not in use, ensure that a certificate has been configured in a local keystore which can validate signatures for this issuer, and the `AuthInfo` stanza of qm.ini configured to access this keystore.
V 10.0.0 112	The queue manager is configured in FIPS mode and validation of the authentication token's signature failed.	Ensure that the queue manager's token authentication key repository contains either the authentication token issuer's public key certificate or the symmetric key that is used to sign the authentication token. The **`CertLabel`** attribute in the AuthToken stanza of the `qm.ini` file must reference the label of the certificate or symmetric key. Issue the **`REFRESH SECURITY TYPE(CONNAUTH)`** MQSC command to bring changes to the queue manager's token authentication configuration into effect. Ensure that the token's signature algorithm used is FIPS compliant. Ensure that the secret used to validate the token's signature is FIPS compliant.

Table 27. Error codes for token authentication configuration errors

Error code (decimal)	Explanation	Response
201	The queue manager is not configured for token authentication, or there is an error in the queue manager token authentication configuration. This error might be caused by one the following problems: • The **AuthToken** stanza is not specified in the `qm.ini` file. • The **AuthToken** stanza is specified more than once in the `qm.ini` file. • A required attribute is not specified in the **AuthToken** stanza. • An invalid value is specified for an attribute in the **AuthToken** stanza.	Specify valid token authentication configuration in the **AuthToken** stanza of the `qm.ini` file. For more information about the **AuthToken** stanza, see AuthToken stanza of the qm.ini file. Issue the **REFRESH SECURITY TYPE(CONNAUTH)** MQSC command to bring changes to the queue manager's token authentication configuration into effect.
202	The connection authentication configuration is incompatible with token authentication. For example, the queue manager is configured for LDAP authentication.	Ensure that the queue manager connection authentication configuration is compatible with token authentication. The AUTHINFO object that is referenced by the queue manager's **CONNAUTH** attribute must be of type `IDPWOS`. Issue the **REFRESH SECURITY TYPE(CONNAUTH)** MQSC command to bring changes to the queue manager's connection authentication configuration into effect.

Table 27. Error codes for token authentication configuration errors (continued)

Error code (decimal)	Explanation	Response
203	An error occurred when accessing the key repository specified by the **KeyStore** attribute in the **AuthToken** stanza of the qm.ini file.	Ensure that the **KeyStore** attribute of the **AuthToken** stanza specifies the path to a valid CMS or PKCS#12 key repository file, and that the mqm user has read permissions for the file. For more information about setting access permissions for the key repository file, see Accessing and securing your key database files on AIX and Linux systems. The queue manager must be able to access the key repository password in one of the following locations: • A file that contains the password that is encrypted by using the **runqmcred** command. The path to the file that contains the encrypted password must be specified by using the **KeyStorePwdFile** attribute in the **AuthToken** stanza of the qm.ini file. • A stash file that is created at the same time as the key repository. If a stash file is used, it must be in the same directory as the key repository, have the same name as the key repository, and have the file extension .sth. If the **KeyStorePwdFile** attribute is specified in the **AuthToken** stanza, it must specify the path to a file that contains a valid encrypted password. The stash file is not used if this attribute is specified.
204	A certificate that is specified with the **CertLabel** attribute in the **AuthToken** stanza of the qm.ini file cannot be loaded from the key repository.	Ensure that each **CertLabel** attribute in the **AuthToken** stanza of the qm.ini file references the correct label of a certificate or symmetric key in the key repository that is specified by using the **KeyStore** attribute. For more information about the **AuthToken** stanza that is used to configure token authentication, see AuthToken stanza of the qm.ini file.
205	The queue manager's security policy configuration, which is specified by using the **SecurityPolicy** attribute in the **Service** stanza of the qm.ini file, is incompatible with token authentication. When the queue manager is configured for token authentication, the security policy is automatically set to UserExternal. However, the security policy cannot be changed if the **SecurityPolicy** attribute in the **Service** stanza of the qm.ini file is explicitly set to group.	Remove the **SecurityPolicy** attribute from the **Service** stanza of the qm.ini file. For more information about the **SecurityPolicy** attribute, see SecurityPolicy.

Table 27. Error codes for token authentication configuration errors (continued)

Error code (decimal)	Explanation	Response
206	The queue manager's connection authentication configuration is set to adopt the context of authenticated users as the context for the connected application. However, the **UserClaim** attribute that specifies the claim that contains the user ID in the authentication token is not specified in the **AuthToken** stanza of the `qm.ini` file.	To adopt a user that is specified in a claim in an authentication token as the context for the connected application, specify the name of the claim that contains the user ID to adopt in the **UserClaim** attribute in the **AuthToken** stanza of the `qm.ini` file. For more information about the **UserClaim** attribute, see UserClaim. If you do not want to adopt a user that is specified in credentials that are supplied by a connecting application as the context for the application, alter the AUTHINFO object that is referenced by the queue manager's CONNAUTH attribute to specify `ADOPTCTX(NO)`. Issue the **REFRESH SECURITY TYPE(CONNAUTH)** MQSC command to bring changes to the queue manager's token authentication and connection authentication configuration into effect.
207	The **CertLabel** attribute in the **AuthToken** stanza of the `qm.ini` file specifies more than the maximum number of certificate labels. Up to 32 certificate labels can be specified with the **CertLabel** attribute.	Specify up to 32 certificate labels to be used for token authentication with the **CertLabel** attribute in the **AuthToken** stanza of the `qm.ini` file. Issue the **REFRESH SECURITY TYPE(CONNAUTH)** MQSC command to bring changes to the queue manager's token authentication configuration into effect.
208	The file containing the encrypted key repository password, which is specified with the **KeyStorePwdFile** attribute in the **AuthToken** stanza of the `qm.ini` file, cannot be read, or the contents of the file are invalid.	Ensure that the key repository password file specified with the **KeyStorePwdFile** attribute in the **AuthToken** stanza of the `qm.ini` file exists. The mqm user must have read permissions for the file, and the file must contain a single line of text that is the encrypted password output by the **runqmcred** command.
209	An error occurred when accessing the stash for the key repository specified by the **KeyStore** attribute in the **AuthToken** stanza of the qm.ini file.	Ensure that a stash file is present for the key repository specified by the **KeyStore** attribute. It must be in the same directory as the key repository, have the same name as the key repository, and have the file extension `.sth`.

Table 28. Error codes for an unexpected token authentication error		
Error code (decimal)	**Explanation**	**Response**
301	An unexpected error occurred when processing the authentication token.	Check for preceding error messages in the queue manager error log. Check for an FDC file that contains more detail about the problem. If the problem cannot be resolved, contact IBM Support.

Table 29. Error codes for JWKS errors		
Error code (decimal)	**Explanation**	**Response**
401	Data returned from the JWKS endpoint does not constitute valid JSON.	Ensure that the JWKS endpoint path is correct and that the data returned is a valid JSON document containing JWKS key data.
402	The remote file/URL was not found (HTTP 404 response)	Ensure that the remote file that is pointed to by the JWKS Endpoint URL is present and that the Endpoint URL is valid.
403	Certificate verification of the peer has failed during the SSL/TLS handshake.	Check for the AMQ5788 error message in the queue manager error log. This error message contains the reason for the certificate verification failure. Correct the error.
404	Host verification has failed - the certificate does not belong to the specified host.	Ensure that the certificate presented by the authentication server correctly includes the hostname as configured in the endpoint URL.
405	Generic communication error with the remote host.	Ensure communication with the remote host is possible. Check for preceding error messages in the queue manager error log. Check for an FDC file that contains more detail about the problem. If the problem cannot be resolved, contact IBM Support.
406	Failed initializing the secure communication library.	Review previous errors and contact IBM Support for assistance if required.
407	Missing one or more of the required attributes in the retrieved JWKS payload. The required attributes are **use**, **kid**, **n**, and **e**, plus V 10.0.0 **kty** (in IBM MQ 9.4.4 and later) or **alg** (in earlier IBM MQ 9.4 releases).	Check that the JWKS payload contains one or more of the required attributes listed.
408	More than 32 certificates were returned in the JWKS payload. Any additional certificates after the first 32 certificates are ignored.	Ensure the JWKS payload contains no more than 32 certificates.
409	No certificates were returned in the JWKS payload.	Ensure the JWKS payload contains the certificates required to validate the JWT token supplied by the client.

Table 29. Error codes for JWKS errors (continued)		
Error code (decimal)	**Explanation**	**Response**
410	The JWKS endpoint SSL certificate revocation status is unknown.	Ensure the system hosting the revocation endpoint specified in the certificate is accessible by the queue manager.
V 10.0.0 411	The queue manager is configured in FIPS mode and certificate verification of the peer has failed during the SSL/TLS handshake.	Check for the AMQ5788 error message in the queue manager error log. This error message contains the reason for the certificate verification failure. Correct the error.
V 10.0.0 412	The queue manager is configured in FIPS mode and host verification has failed - the certificate does not belong to the specified host.	Ensure that the certificate presented by the authentication server correctly includes the hostname as configured in the endpoint URL.
V 10.0.0 413	The queue manager is configured in FIPS mode and a generic communication error occurred with the remote host.	Ensure communication with the remote host is possible. Check for preceding error messages in the queue manager error log. Check for an FDC file that contains more detail about the problem. If the problem cannot be resolved, contact IBM Support.
V 10.0.0 414	The queue manager is configured in FIPS mode and failed initializing the secure communication library.	Review previous errors and contact IBM Support for assistance if required.
V 10.0.0 415	A timeout occurred when the queue manager tried to contact the JWKS endpoint to obtain an update of token issuer information.	Check for preceding error messages in the queue manager error log; the problem might be caused by a configuration error. Check the connectivity between your queue manager and the authentication server.

Table 30. Error codes for HTTPS errors		
Error code (decimal)	**Explanation**	**Response**
501	The HTTPS Key store file is not accessible.	Ensure that an HTTPS Key store file exists at the default location `QMdata/SSL/mqdefcer.p12` or another location specified by **HTTPSKeyStore** in the SSL Stanza of the qm.ini file. Confirm that the permissions of this file permit read access to the `mqm` user id.
502	The HTTPS Stash file is not accessible.	Ensure that an HTTPS Key store stash file exists at the default location `[QM data dir]/SSL/mqdefcer.sth` or equivalent if specified by **HTTPSKeyStore** in the SSL Stanza of the qm.ini file. Confirm that the permissions of this file permit read access to the `mqm` user ID.

Related concepts
Working with authentication tokens
Troubleshooting authentication token problems
Related tasks
Using authentication tokens in an application
Configuring a queue manager to accept **AuthTokens**
Related reference
AuthToken stanza of the `qm.ini` file

HTTP error codes

Use the codes and explanations to understand and correct errors that can be caused when accessing HTTPS resources from the IBM MQ client library (for example, CCDTURL).

These tables document error codes, in decimal form, that are displayed in message AMQ9795E.

Table 31. Error codes for HTTP errors

Error code (decimal)	Explanation	Response
101	The URL that was specified is too long.	The URL that was provided for a remote resource (for example, the CCDTURL property) exceeds the maximum path length for this operating system.
102	HTTPS initialization failed.	There was a problem initializing the HTTP communication library. This error is usually accompanied by other error messages that indicate the root cause. If required, contact IBM support.
103	There was a communication error with the remote host.	See additional information in the message to diagnose and correct the failure.
104	The remote file or URL was not found.	See additional information in the message to diagnose and correct the failure.
105	The certificate does not belong to the specified host.	Ensure that the distinguished name in the HTTP server's certificate matches the hostname that is specified. If you are intentionally using a certificate that does not match, you can reduce the level of verification by using the MQSCO structure or the `mqclient.ini` file to reduce the certificate validation level to ANY.
106	The endpoint certificate revocation status is unknown.	Ensure that the CRL or OCSP server that is specified in your HTTP server's certificate is valid and can be reached from the client host. If certificate revocation checks are not required, you can disable them by using the MQSCO HTTPSCertRevocation field or the HTTPSCertRevocation attribute in the `mqclient.ini` file.
107	There was a problem with the HTTPS keystore.	See additional information in the message to diagnose and correct the failure.

Table 31. Error codes for HTTP errors (continued)		
Error code (decimal)	**Explanation**	**Response**
108	The HTTPS keystore stash file is not accessible.	The HTTPSKeyStore must be accompanied by a stash file (with an extension of `.sth`) in the same directory, with the same name as the keystore file. Ensure that the stash file exists and is accessible (has `read` permission) to the user ID that is running the IBM MQ client application.

Related reference

MQSCO - SSL/TLS configuration options

SSL stanza of the client configuration file

Transport Layer Security (TLS) return codes

IBM MQ can use TLS with the various communication protocols. Use this topic to identify the error codes that can be returned by TLS.

The table in this appendix documents the return codes, in decimal form, from the TLS that can be returned in messages from the distributed queuing component.

Table 32. TLS return codes	
Return code (decimal)	**Explanation**
1	Handle is not valid.
3	An internal error has occurred.
4	Insufficient storage is available
5	Handle is in the incorrect state.
6	Key label is not found.
7	No certificates available.
8	Certificate validation error.
9	Cryptographic processing error.
10	ASN processing error.
11	LDAP processing error.
12	An unexpected error has occurred.
102	Error detected while reading key database or SAF key ring.
103	Incorrect key database record format.
106	Incorrect key database password.
109	No certificate authority certificates.
201	No key database password supplied.
202	Error detected while opening the key database.

Table 32. TLS return codes (continued)

Return code (decimal)	Explanation
203	Unable to generate temporary key pair
204	Key database password is expired.
302	Connection is active.
401	Certificate is expired or is not valid yet.
402	No TLS cipher specifications.
403	No certificate received from partner.
404	The certificate is bad.
405	Certificate format is not supported.
406	Error while reading or writing data.
407	Key label does not exist.
408	Key database password is not correct.
410	TLS message format is incorrect.
411	Message authentication code is incorrect.
412	TLS protocol or certificate type is not supported.
413	Certificate signature is incorrect.
414	Certificate is not valid.
415	TLS protocol violation.
416	Permission denied.
417	Self-signed certificate cannot be validated.
420	Socket closed by remote partner.
421	SSL 2.0 cipher is not valid.
422	SSL 3.0 cipher is not valid.
427	LDAP is not available.
428	Key entry does not contain a private key.
429	SSL 2.0 header is not valid.
431	Certificate is revoked.
432	Session renegotiation is not allowed.
433	Key exceeds allowable export size.
434	Certificate key is not compatible with cipher suite.
435	Certificate authority is unknown.
436	Certificate revocation list cannot be processed.
437	Connection closed.
438	Internal error reported by remote partner.
439	Unknown alert received from remote partner.

Table 32. TLS return codes (continued)	
Return code (decimal)	**Explanation**
501	Buffer size is not valid.
502	Socket request would block.
503	Socket read request would block.
504	Socket write request would block.
505	Record overflow.
601	Protocol is not TLS 1.
602	Function identifier is not valid.
701	Attribute identifier is not valid.
702	The attribute has a negative length, which is invalid.
703	The enumeration value is invalid for the specified enumeration type.
704	Invalid parameter list for replacing the SID cache routines.
705	The value is not a valid number.
706	Conflicting parameters were set for additional certificate validation
707	The AES cryptographic algorithm is not supported.
708	The PEERID does not have the correct length.
1501	GSK_SC_OK
1502	GSK_SC_CANCEL
1601	The trace started successfully.
1602	The trace stopped successfully.
1603	No trace file was previously started so it cannot be stopped.
1604	Trace file already started so it cannot be started again.
1605	Trace file cannot be opened. The first parameter of gsk_start_trace() must be a valid full path filename.

In some cases, the secure sockets library reports a certificate validation error in an AMQ9633 error message. Table 2 lists the certificate validation errors that can be returned in messages from the distributed queuing component.

Table 33. Certificate validation errors. A table listing return codes and explanations for certificate validation errors that can be returned in messages from the distributed queuing component.	
Return code (decimal)	**Explanation**
575001	Internal error
575002	ASN error due to a malformed certificate
575003	Cryptographic error
575004	Key database error

Table 33. Certificate validation errors.

A table listing return codes and explanations for certificate validation errors that can be returned in messages from the distributed queuing component.

(continued)

Return code (decimal)	Explanation
575005	Directory error
575006	Invalid implementation library
575008	No appropriate validator
575009	The root CA is not trusted
575010	No certificate chain was built
575011	Digital signature algorithm mismatch
575012	Digital signature mismatch
575013	X.509 version does not allow Key IDs
575014	X.509 version does not allow extensions
575015	Unknown X.509 certificate version
575016	The certificate validity range is invalid
575017	The certificate is not yet valid
575018	The certificate has expired
575019	The certificate contains unknown critical extensions
575020	The certificate contains duplicate extensions
575021	The issuers directory name does not match the issuer's issuer
575022	The Authority Key ID serial number value does not match the serial number of the issuer
575023	The Authority Key ID and Subject Key ID do not match
575024	Unrecognized issuer alternative name
575025	The certificate Basic Constraints forbid use as a CA
575026	The certificate has a non-zero Basic Constraints path length but is not a CA
575027	The certificate Basic Constraints maximum path length was exceeded
575028	The certificate is not permitted to sign other certificates
575029	The certificate is not signed by a CA
575030	Unrecognized Subject Alternative Name
575031	The certificate chain is invalid
575032	The certificate is revoked
575033	Unrecognized CRL distribution point
575034	Name chaining failed
575035	Certificate is not in a chain
575036	The CRL is not yet valid

Table 33. Certificate validation errors.

A table listing return codes and explanations for certificate validation errors that can be returned in messages from the distributed queuing component.

(continued)

Return code (decimal)	**Explanation**
575037	The CRL has expired
575038	The certificate version does not allow critical extensions
575039	Unknown CRL distribution points
575040	No CRLs for CRL distribution points
575041	Indirect CRLs are not supported
575042	Missing issuing CRL distribution point name
575043	Distribution points do not match
575044	No available CRL data source
575045	CA Subject name is null
575046	Distinguished names do not chain
575047	Missing Subject Alternative Name
575048	Unique ID mismatch
575049	Name not permitted
575050	Name excluded
575051	CA certificate is missing Critical Basic Constraints
575052	Name constraints are not critical
575053	Name constraints minimum subtree value if set is not zero
575054	Name constraints maximum subtree value if set is not allowed
575055	Unsupported name constraint
575056	Empty policy constraints
575057	Bad certificate policies
575058	Certificate policies not acceptable
575059	Bad acceptable certificate policies
575060	Certificate policy mappings are critical
575061	Revocation status could not be determined
575062	Extended key usage error
575063	Unknown OCSP version
575064	Unknown OCSP response
575065	Bad OCSP key usage extension
575066	Bad OCSP nonce
575067	Missing OCSP nonce

Table 33. Certificate validation errors.

A table listing return codes and explanations for certificate validation errors that can be returned in messages from the distributed queuing component.

(continued)

Return code (decimal)	Explanation
575068	No OCSP client available
575069	Policy not critical
575070	OCSP old but good
575071	OCSP old but revoked
575072	Incorrect curve
575073	Incorrect key size
575074	Incorrect signature algorithm

A possible explanation for an error message being issued with return code 575074 is that is that the sole CipherSpec chosen by the client demands the use of a server certificate with an Elliptic curve signature, but the server certificate was using an RSA signature. The reverse could be true, that is, the server has an elliptic curve certificate but the sole CipherSpec proposed was RSA.

Related concepts

"IBM MQ for z/OS messages, completion, and reason codes" on page 289
Use this topic to interpret and understand the messages and codes issued by IBM MQ for z/OS.

Related reference

"AMQ messages on Multiplatforms" on page 244
The AMQ series of diagnostic messages are listed in this section in numerical order, grouped according to the part of IBM MQ from which they originate.

"API completion and reason codes" on page 1151
For each call, a completion code and a reason code are returned by the queue manager or by an exit routine, to indicate the success or failure of the call.

"PCF reason codes" on page 1360
Reason codes might be returned by a broker in response to a command message in PCF format, depending on the parameters used in that message.

"WCF custom channel exceptions" on page 1446
Diagnostic messages are listed in this topic in numeric order, grouped according to the part of the Microsoft Windows Communication Foundation (WCF) custom channel from which they originate.

WCF custom channel exceptions

Diagnostic messages are listed in this topic in numeric order, grouped according to the part of the Microsoft Windows Communication Foundation (WCF) custom channel from which they originate.

Reading a message

For each message, this information is provided:

- The message identifier, in two parts:
  1. The characters "WCFCH" which identify the message as being from the Microsoft Windows Communication Foundation (WCF) custom channel for IBM MQ
  2. A four-digit decimal code followed by the character 'E'

- The text of the message.
- An explanation of the message giving further information.
- The response required from the user. In some cases, particularly for information messages, the response required might be "none".

Message variables

Some messages display text or numbers that vary according to the circumstances causing the message to occur; these circumstances are known as *message variables*. The message variables are indicated as {0}, {1}, and so on.

In some cases a message might have variables in the Explanation or Response. Find the values of the message variables by looking in the error log. The complete message, including the Explanation and the Response, is recorded there.

The following message types are described:

"WCFCH0001E-0100E: General/State messages" on page 1447
"WCFCH0101E-0200E: URI Properties messages" on page 1449
"WCFCH0201E-0300E: Factory/Listener messages" on page 1450
"WCFCH0301E-0400E: Channel messages" on page 1451
"WCFCH0401E-0500E: Binding messages" on page 1452
"WCFCH0501E-0600E: Binding properties messages" on page 1454
"WCFCH0601E-0700E: Async operations messages" on page 1454

Related concepts

"IBM MQ for z/OS messages, completion, and reason codes" on page 289

Use this topic to interpret and understand the messages and codes issued by IBM MQ for z/OS.

Related reference

"AMQ messages on Multiplatforms" on page 244

The AMQ series of diagnostic messages are listed in this section in numerical order, grouped according to the part of IBM MQ from which they originate.

"API completion and reason codes" on page 1151

For each call, a completion code and a reason code are returned by the queue manager or by an exit routine, to indicate the success or failure of the call.

"PCF reason codes" on page 1360

Reason codes might be returned by a broker in response to a command message in PCF format, depending on the parameters used in that message.

"Transport Layer Security (TLS) return codes" on page 1441

IBM MQ can use TLS with the various communication protocols. Use this topic to identify the error codes that can be returned by TLS.

"WCF custom channel exceptions" on page 1446

Diagnostic messages are listed in this topic in numeric order, grouped according to the part of the Microsoft Windows Communication Foundation (WCF) custom channel from which they originate.

WCFCH0001E-0100E: General/State messages

Use the following information to understand WCFCH0001E-0100E general/state messages.

WCFCH0001E
An object cannot be opened because its state is '{0}'.

Explanation
An internal error has occurred.

Response
Use the standard facilities supplied with your system to record the problem identifier and to save any generated output files. Use either the IBM MQ support web page, or the IBM Support Assistant web

page, to see if a solution is already available. If you cannot find a match, contact IBM Support. Do not discard these files until the problem has been resolved.

WCFCH0002E
An object cannot be closed because its state is '{0}'.

Explanation
An internal error has occurred.

Response
Use the standard facilities supplied with your system to record the problem identifier and to save any generated output files. Use either the IBM MQ support web page, or the IBM Support Assistant web page, to see if a solution is already available. If you cannot find a match, contact IBM Support. Do not discard these files until the problem has been resolved.

WCFCH0003E
An object cannot be used because its state is '{0}'.

Explanation
An internal error has occurred.

Response
Use the standard facilities supplied with your system to record the problem identifier and to save any generated output files. Use either the IBM MQ support web page, or the IBM Support Assistant web page, to see if a solution is already available. If you cannot find a match, contact IBM Support. Do not discard these files until the problem has been resolved.

WCFCH0004E
The specified 'Timeout' value '{0}' is out of range.

Explanation
The value is out of range, it must be greater than or equal to 'TimeSpan.Zero'.

Response
Specify a value which is in range or, to disable Timeout, specify a 'TimeSpan.MaxValue' value.

WCFCH0005E
The operation did not complete within the specified time of '{0}' for endpoint address '{1}'.

Explanation
A timeout occurred.

Response
Investigate the cause for the timeout.

WCFCH0006E
The parameter '{0}' is not of the expected type '{1}'

Explanation
A parameter with an unexpected type has been passed to a method call.

Response
Review the exception stack trace for further information.

WCFCH0007E
The parameter '{0}' must not be null.

Explanation
A method has been called with a required parameter set to a null value.

Response
Modify the application to provide a value for this parameter.

WCFCH0008E
An error occurred while processing an operation for endpoint address '{0}'.

Explanation
The operation failed to complete.

Response
Review the linked exceptions and stack trace for further information.

WCFCH0101E-0200E: URI Properties messages

Use the following information to understand WCFCH0101E-0200E URI properties messages.

WCFCH0101E
The endpoint URI must start with the valid character string '{0}'.

Explanation
The endpoint URI is incorrect, it must start with a valid character string.

Response
Specify an endpoint URI which starts with a valid character string.

WCFCH0102E
The endpoint URI must contain a '{0}' parameter with a value.

Explanation
The endpoint URI is incorrect, a parameter and its value are missing.

Response
Specify an endpoint URI with a value for this parameter.

WCFCH0103E
The endpoint URI must contain a '{0}' parameter with a value of '{1}'.

Explanation
The endpoint URI is incorrect, the parameter must contain the correct value.

Response
Specify an endpoint URI with a correct parameter and value.

WCFCH0104E
The endpoint URI contains a '{0}' parameter with an invalid value of '{1}'.

Explanation
The endpoint URI is incorrect, a valid parameter value must be specified.

Response
Specify an endpoint URI with a correct value for this parameter.

WCFCH0105E
The endpoint URI contains a '{0}' parameter with an invalid queue or queue manager name.

Explanation
The endpoint URI is incorrect, a valid queue and queue manager name must be specified.

Response
Specify an endpoint URI with valid values for the queue and the queue manager.

WCFCH0106E
The '{0}' property is a required property and must appear as the first property in the endpoint URI.

Explanation
The endpoint URI is incorrect, a parameter is either missing or in the wrong position.

Response
Specify an endpoint URI which contains this property as the first parameter.

WCFCH0107E
The property '{1}' cannot be used when the binding property is set to '{0}'.

Explanation
The endpoint URI connectionFactory parameter is incorrect, an invalid combination of properties has been used.

Response
Specify an endpoint URI connectionFactory which contains a valid combination of properties or binding.

WCFCH0109E
Property '{1}' must also be specified when property '{0}' is specified.

Explanation
The endpoint URI connectionFactory parameter is incorrect, it contains an invalid combination of properties.

Response
Specify an endpoint URI connectionFactory which contains a valid combination of properties.

WCFCH0110E
Property '{0}' has an invalid value '{1}'.

Explanation
The endpoint URI connectionFactory parameter is incorrect, the property does not contain a valid value.

Response
Specify an endpoint URI connectionFactory which contains a valid value for the property.

WCFCH0111E
The value '{0}' is not supported for the binding mode property. XA operations are not supported.

Explanation
The endpoint URI connectionFactory parameter is incorrect, the binding mode is not supported.

Response
Specify an endpoint URI connectionFactory which contains a valid value for the binding mode.

WCFCH0112E
The endpoint URI '{0}' is badly formatted.

Explanation
The endpoint URI must follow the format described in the documentation.

Response
Review the endpoint URI to ensure that it contains a valid value.

WCFCH0201E-0300E: Factory/Listener messages

Use the following information to understand WCFCH0201E-0300E factory/listener messages.

WCFCH0201E
Channel shape '{0}' is not supported.

Explanation
The users application or the WCF service contract has requested a channel shape which is not supported.

Response
Identify and use a channel shape which is supported by the channel.

WCFCH0202E
'{0}' MessageEncodingBindingElements have been specified.

Explanation
The WCF binding configuration used by an application contains more than one message encoder.

Response
Specify no more then 1 MessageEncodingBindingElement in the binding configuration.

WCFCH0203E
The endpoint URI address for the service listener must be used exactly as provided.

Explanation
The binding information for the endpoint URI address must specify a value of 'Explicit' for the 'listenUriMode' parameter.

Response
Change the parameter value to 'Explicit'.

WCFCH0204E
TLS is not supported for managed client connections [endpoint URI: '{0}'].

Explanation
The endpoint URI specifies a TLS connection type which is only supported for unmanaged client connections.

Response
Modify the channels binding properties to specify an unmanaged client connection mode.

Related concepts
WCF custom channel shapes

WCFCH0301E-0400E: Channel messages

Use the following information to understand WCFCH0301E-0400E channel messages.

WCFCH0301E
The URI scheme '{0}' is not supported.

Explanation
The requested endpoint contains a URI scheme which is not supported by the channel.

Response
Specify a valid scheme for the channel.

WCFCH0302E
The received message '{0}' was not a JMS bytes or a JMS text message.

Explanation
A message has been received but it is not of the correct type. It must be either a JMS bytes message or a JMS text message.

Response
Check the origin and contents of the message and determine the cause for it being incorrect.

WCFCH0303E
'ReplyTo' destination missing.

Explanation
A reply cannot be sent because the original request does not contain a 'ReplyTo' destination.

Response
Investigate the reason for the missing destination value.

WCFCH0304E
The connection attempt to queue manager '{0}' failed for endpoint '{1}'

Explanation
The queue manager could not be contacted at the given address.

Response
Review the linked exception for further details.

WCFCH0305E
The connection attempt to the default queue manager failed for endpoint '{0}'

Explanation
The queue manager could not be contacted at the given address.

Response
Review the linked exception for further details.

WCFCH0306E
An error occurred while attempting to receive data from endpoint '{0}'

Explanation
The operation could not be completed.

Response
Review the linked exception for further details.

WCFCH0307E
An error occurred while attempting to send data for endpoint '{0}'

Explanation
The operation could not be completed.

Response
Review the linked exception for further details.

WCFCH0308E
An error occurred while attempting to close the channel for endpoint '{0}'

Explanation
The operation could not be completed.

Response
Review the linked exception for further details.

WCFCH0309E
An error occurred while attempting to open the channel for endpoint '{0}'

Explanation
The operation could not be completed.

Response
The endpoint might be down, unavailable, or unreachable, review the linked exception for further details.

WCFCH0310E
The timeout '{0}' was exceeded while attempting to receive data from endpoint '{0}'

Explanation
The operation did not complete in the time allowed.

Response
Review the system status and configuration and increase the timeout if required.

WCFCH0311E
The timeout '{0}' was exceeded while attempting to send data for endpoint '{0}'

Explanation
The operation did not complete in the time allowed.

Response
Review the system status and configuration and increase the timeout if required.

WCFCH0312E
The timeout '{0}' was exceeded while attempting to close the channel for endpoint '{0}'

Explanation
The operation did not complete in the time allowed.

Response
Review the system status and configuration and increase the timeout if required.

WCFCH0313E
The timeout '{0}' was exceeded while attempting to open the channel for endpoint '{0}'

Explanation
The operation did not complete in the time allowed.

Response
The endpoint might be down, unavailable, or unreachable, review the system status and configuration and increase the timeout if required.

WCFCH0401E-0500E: Binding messages

Use the following information to understand WCFCH0401E-0500E binding messages.

WCFCH0401E
No context.

Explanation
An internal error has occurred.

Response
Use the standard facilities supplied with your system to record the problem identifier and to save any generated output files. Use either the IBM Support Portal for IBM MQ (see https://www.ibm.com/support/entry/portal/Overview/Software/WebSphere/WebSphere_MQ), or the IBM Support Assistant (at https://www.ibm.com/support/home/product/C100515X13178X21/other_software/ibm_support_assistant), to see if a solution is already available. If you cannot find a match, contact IBM Support. Do not discard these files until the problem has been resolved.

WCFCH0402E
Channel type '{0}' is not supported.

Explanation
The users application or the WCF service contract has requested a channel shape which is not supported.

Response
Identify and use a channel shape which is supported by the channel.

WCFCH0403E
No exporter.

Explanation
An internal error has occurred.

Response
Use the standard facilities supplied with your system to record the problem identifier and to save any generated output files. Use either the IBM Support Portal for IBM MQ (see https://www.ibm.com/support/entry/portal/Overview/Software/WebSphere/WebSphere_MQ), or the IBM Support Assistant (at https://www.ibm.com/support/home/product/C100515X13178X21/other_software/ibm_support_assistant), to see if a solution is already available. If you cannot find a match, contact IBM Support. Do not discard these files until the problem has been resolved.

WCFCH0404E
The WS-Addressing version '{0}' is not supported.

Explanation
The addressing version specified is not supported.

Response
Specify an addressing version which is supported.

WCFCH0405E
No importer.

Explanation
An internal error has occurred.

Response
Use the standard facilities supplied with your system to record the problem identifier and to save any generated output files. Use either the IBM Support Portal for IBM MQ (see https://www.ibm.com/support/entry/portal/Overview/Software/WebSphere/WebSphere_MQ), or the IBM Support Assistant (at https://www.ibm.com/support/home/product/C100515X13178X21/other_software/ibm_support_assistant), to see if a solution is already available. If you cannot find a match, contact IBM Support. Do not discard these files until the problem has been resolved.

WCFCH0406E
Endpoint 'Binding' value missing.

Explanation
An internal error has occurred.

Response
Use the standard facilities supplied with your system to record the problem identifier and to save any generated output files. Use either the IBM Support Portal for IBM MQ

(see https://www.ibm.com/support/entry/portal/Overview/Software/WebSphere/WebSphere_MQ), or the IBM Support Assistant (at https://www.ibm.com/support/home/product/C100515X13178X21/other_software/ibm_support_assistant), to see if a solution is already available. If you cannot find a match, contact IBM Support. Do not discard these files until the problem has been resolved.

Related concepts

WCF custom channel shapes

WCFCH0501E-0600E: Binding properties messages

Use the following information to understand WCFCH0501E-0600E binding properties messages.

WCFCH0501E

The binding property '{0}' has an invalid value '{1}'.

Explanation

An invalid value has been specified for a binding property.

Response

Specify a valid value for the property.

WCFCH0601E-0700E: Async operations messages

Use the following information to understand WCFCH0601E-0700E async operations messages.

WCFCH0601E

The async result parameter '{0}' object is not valid for this call.

Explanation

An invalid async result object has been provided.

Response

Specify a valid value for the parameter.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information 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 document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement 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 publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Software Interoperability Coordinator, Department 49XA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

© Copyright IBM Corp. 2007, 2026

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 Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments 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 contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

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.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming interface information

Programming interface information, if provided, is intended to help you create application software for use with this program.

This book contains information on intended programming interfaces that allow the customer to write programs to obtain the services of IBM MQ.

However, this information may also contain diagnosis, modification, and tuning information. Diagnosis, modification and tuning information is provided to help you debug your application software.

Important: Do not use this diagnosis, modification, and tuning information as a programming interface because it is subject to change.

Trademarks

IBM, the IBM logo, ibm.com®, are trademarks of IBM Corporation, registered in many jurisdictions worldwide. A current list of IBM trademarks is available on the Web at "Copyright and trademark information"www.ibm.com/legal/copytrade.shtml. Other product and service names might be trademarks of IBM or other companies.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

This product includes software developed by the Eclipse Project (https://www.eclipse.org/).

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Part Number:

(1P) P/N: