

9.0

IBM MQ Explorer

IBM

Note

Before using this information and the product it supports, read the information in [“Notices” on page 665](#).

This edition applies to version 9 release 0 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, 2022.**

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

Contents

IBM MQ Explorer.....	5
IBM MQ Explorer Requirements.....	18
What's new and what's changed in IBM MQ Explorer.....	19
Launching IBM MQ Explorer.....	23
Multiple installations of IBM MQ Explorer.....	23
Installing into Eclipse environments.....	24
Displaying context-sensitive help (pop-up window help).....	25
Configuring IBM MQ using IBM MQ Explorer.....	26
Creating and configuring queue managers and objects.....	27
Testing your object definitions for problems.....	60
Sending test messages.....	89
Starting and stopping objects and services.....	91
Showing or hiding a queue manager.....	98
Connecting or disconnecting a queue manager.....	109
Administering remote queue managers.....	111
Maintaining intercommunications along message channels.....	113
Configuring publish/subscribe messaging.....	117
Configuring publish/subscribe for IBM WebSphere MQ 7.0 and later queue managers.....	129
Managing multi-instance queue managers.....	140
Creating and configuring a queue manager cluster.....	144
Managing security and authorities.....	155
Viewing the status of objects.....	202
Viewing and closing connections to applications.....	203
Creating and configuring JMS administered objects.....	204
JMS contexts.....	205
JMS connection factories.....	206
JMS destinations (queues and topics).....	208
Messaging providers for IBM MQ classes for JMS.....	209
Adding an initial context.....	209
Connecting and disconnecting an initial context.....	211
Removing an initial context.....	212
Creating a connection factory.....	213
Creating a destination.....	215
Creating a JMS object and an IBM MQ object simultaneously.....	216
Creating a JMS object from an IBM MQ object.....	217
Copying an administered object.....	218
Changing the transport type used for connections.....	219
Creating a subcontext.....	220
Renaming an administered object.....	220
Renaming a context.....	221
Deleting an administered object.....	222
Deleting a subcontext.....	222
Configuring IBM MQ Explorer.....	223
Filtering the objects displayed in tables.....	225
Creating and configuring a service definition.....	228
Creating and configuring a queue manager set.....	233
Define schemes to change the order of columns in tables.....	250
Changing the colors.....	254
Enabling installed plug-ins.....	255
Changing the refresh frequency of queue manager information.....	255
Specifying the default values used to connect to remote queue managers.....	256
Exporting and importing settings.....	258

Including SYSTEM objects when you run tests.....	259
Including hidden queue managers in test configurations.....	259
Displaying object authority settings as text.....	260
Advanced Message Security.....	260
Message signing.....	260
Message encryption.....	261
Distinguished names.....	261
Troubleshooting.....	262
Using IBM MQ Explorer trace.....	262
Using IBM MQ Explorer trace in other Eclipse environments.....	263
Using IBM MQ trace.....	268
MQ Telemetry.....	269
MQ Telemetry objects.....	269
MQTT client utility.....	271
Configuring MQ Telemetry using IBM MQ Explorer.....	275
Administering MQ Telemetry using IBM MQ Explorer.....	279
Troubleshooting MQ Telemetry using IBM MQ Explorer.....	283
MQ Telemetry Reference.....	286
IBM MQ Tutorials.....	289
Tutorial 1: Sending a message to a local queue.....	289
Tutorial 2: Sending a message to a remote queue.....	296
Tutorial 3: Sending a message on a client-server configuration.....	304
Reference.....	309
Accessibility in IBM MQ Explorer.....	309
Icons in IBM MQ Explorer.....	310
Views in IBM MQ Explorer.....	315
Preferences for IBM MQ Explorer.....	324
Properties.....	339
Status attributes.....	604
Byte array dialog.....	655
Strings in property dialogs.....	655
Extending IBM MQ Explorer.....	656
Importing the sample Eclipse plug-ins.....	656
Writing an Eclipse plug-in for IBM MQ Explorer.....	657
Applying plug-ins to IBM MQ Explorer.....	662
API Reference.....	662
Notices.....	665
Programming interface information.....	666
Trademarks.....	666

Introduction to IBM MQ Explorer

IBM MQ Explorer is the graphical user interface in which you can administer and monitor IBM MQ objects, whether they are hosted by your local computer or on a remote system.

IBM MQ Explorer runs on Windows and Linux® x86-64. It can remotely connect to queue managers that are running on any supported platform including z/OS®, enabling your entire messaging backbone to be viewed, explored, and altered from the console.

IBM MQ Explorer is built on open source Eclipse technology. As such, IBM MQ Explorer is highly customizable and fully extensible. You can add new tools as plug-ins to IBM MQ Explorer to provide new features in a way that is integrated into the console.

Related reference

[“Accessibility in IBM MQ Explorer” on page 309](#)

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully.

Entities for MQTT clients

MQTT names for client pack, clients, and client sample apps

- machine-to-machine (M2M)
- Mobile Messaging and M2M
- IBM Messaging Telemetry Clients
- IBM Messaging Telemetry Clients SupportPac
- MQTT client for Java
- MQTT messaging client for JavaScript
- MQTT client for C
- MQTT client sample Java app
- MQTT client sample Java app for Android
- MQTT messaging client sample JavaScript pages
- MQTT client sample C app

MQTT phrases, xrefs, filepaths

The examples are `MQTTV3ASample.c` and `MQTTV3ASSample.c` in `sdkroot\SDK\clients\c\samples`.

MQTT build options for different platforms

Install a C development environment on the platform on which you are building.

[System Requirements for IBM Mobile Messaging and M2M Client Pack MA9B \(superseded\)](#)

The client identifier must be unique across all clients that connect to the server, and must not be the same as the queue manager name on the server.

The configuration gives everyone permission to publish and subscribe to any topic. The configuration of security and access control is minimal and is intended only for a queue manager that is on a secure network with restricted access. To run IBM MQ and MQTT in an insecure environment, you must configure security. To configure security for IBM MQ and MQTT, see the related links at the end of this task.

If you use the default `MqttConnectOptions`, or set `MqttConnectOptions.cleanSession` to `true` before connecting the client, any old subscriptions for the client are removed when the client connects. Any new subscriptions the client makes during the session are removed when it disconnects.

If you set `MqttConnectOptions.cleanSession` to `false` before connecting, any subscriptions the client creates are added to all the subscriptions that existed for the client before it connected. All the subscriptions remain active when the client disconnects.

Another way of understanding the way the `cleanSession` attribute affects subscriptions is to

think of it as a modal attribute. In its default mode, `cleanSession=true`, the client creates subscriptions and receives publications only within the scope of the session. In the alternative mode, `cleanSession=false`, subscriptions are durable. The client can connect and disconnect and its subscriptions remain active. When the client reconnects, it receives any undelivered publications. While it is connected, it can modify the set of subscriptions that are active on its behalf.

You must set the `cleanSession` mode before connecting; the mode lasts for the whole session. To change its setting, you must disconnect and reconnect the client. If you change modes from using `cleanSession=false` to `cleanSession=true`, all previous subscriptions for the client, and any publications that have not been received, are discarded.

Example scripts to configure SSL certificates for Windows

Change `sdkroot` to suit your environment.

Install a Java development kit (JDK) at Version 7 or later.

Version 7 is required to run the **keytool** command to certify certificates. If you are not going to certify certificates, you do not require the Version 7 JDK.

Build output is produced in the current directory.

MQTTCLIENT_DIR must point to the base directory containing the MQTT client source code.

Default MQTTCLIENT_DIR is the current directory

Default TOOL_DIR is /Applications/Xcode.app/Contents/Developer/Platforms

Default OPENSSL_DIR is `sdkroot/openssl`, relative to `sdkroot/sdk/clients/c/mqttv3c/src`

OPENSSL_DIR must point to the base directory containing the OpenSSL build.

Example: `make -f MQTTios.mak MQTTCLIENT_DIR=sdkroot/sdk/clients/c/mqttv3c/src` all

ifndef TOOL_DIR

TOOL_DIR = /Applications/Xcode.app/Contents/Developer/Platforms endif

IPHONE_SDK = iPhoneOS.platform/Developer/SDKs/iPhoneOS6.0.sdk

IPHONESIM_SDK = iPhoneSimulator.platform/Developer/SDKs/iPhoneSimulator6.0.sdk

SDK_ARM = \${TOOL_DIR}/\${IPHONE_SDK}

SDK_i386 = \${TOOL_DIR}/\${IPHONESIM_SDK}

MQTTLIB = mqttv3c

SOURCE_FILES = \${filter-out \${MQTTCLIENT_DIR}/MQTTAsync.c \${MQTTCLIENT_DIR}/SSLSocket.c, \${ALL_SOURCE_FILES}}

MQTTLIB_DARWIN = darwin_x86_64/lib\${MQTTLIB}.a

MQTTLIB_S = mqttv3cs

SOURCE_FILES_S = \${filter-out \${MQTTCLIENT_DIR}/MQTTAsync.c, \${ALL_SOURCE_FILES}}

MQTTLIB_DARWIN_S = darwin_x86_64/lib\${MQTTLIB_S}.a

MQTTLIB_A = mqttv3a

SOURCE_FILES_A = \${filter-out \${MQTTCLIENT_DIR}/MQTTClient.c \${MQTTCLIENT_DIR}/SSLSocket.c, \${ALL_SOURCE_FILES}}

MQTTLIB_DARWIN_A = darwin_x86_64/lib\${MQTTLIB_A}.a

MQTTLIB_AS = mqttv3as

SOURCE_FILES_AS = \${filter-out \${MQTTCLIENT_DIR}/MQTTClient.c, \${ALL_SOURCE_FILES}}

MQTTLIB_DARWIN_AS = darwin_x86_64/lib\${MQTTLIB_AS}.a

CC = iPhoneOS.platform/Developer/usr/bin/gcc

CC_armv7 = \${TOOL_DIR}/\${CC} -arch armv7

CC_armv7s = \${TOOL_DIR}/\${CC} -arch armv7s

CC_i386 = \${TOOL_DIR}/\${CC} -arch i386

CCFLAGS = -Os -Wall -fomit-frame-pointer

CCFLAGS_SO_ARM = \${CCFLAGS} -isysroot \${SDK_ARM} -I\${OPENSSL_DIR}/include -L\${SDK_ARM}/usr/lib/system

CCFLAGS_SO_i386 = \${CCFLAGS} -isysroot \${SDK_i386} -I\${OPENSSL_DIR}/include -L\${SDK_i386}/usr/lib/system

all: \${MQTTLIB_DARWIN} \${MQTTLIB_DARWIN_A} \${MQTTLIB_DARWIN_AS} \${MQTTLIB_DARWIN_S} \${MQTTLIB_DARWIN}: \${SOURCE_FILES}

-mkdir darwin_x86_64

\${CC_armv7} \${CCFLAGS_SO_ARM} -c \${SOURCE_FILES} -DUSE_NAMED_SEMAPHORES

-DNOSIGPIPE

libtool -static -syslibroot \${SDK_ARM} -o \${@}.armv7 *.o

rm *.o

\${CC_armv7s} \${CCFLAGS_SO_ARM} -c \${SOURCE_FILES} -DUSE_NAMED_SEMAPHORES

```

-DNOSIGPIPE
libtool -static -syslibroot ${SDK_ARM} -o ${@.armv7s *.o}
rm *.o
${CC_i386} ${CCFLAGS_SO_i386} -c ${SOURCE_FILES} -DUSE_NAMED_SEMAPHORES -DNOSIGPIPE
libtool -static -syslibroot ${SDK_i386} -o ${@.i386 *.o}
rm *.o
lipo -create ${@.armv7} ${@.armv7s} ${@.i386} -output ${@}
${MQTTLIB_DARWIN_A}: ${SOURCE_FILES_A}
-mkdir darwin_x86_64
${CC_armv7} ${CCFLAGS_SO_ARM} -c ${SOURCE_FILES_A} -DMQTT_ASYNC
-DUSE_NAMED_SEMAPHORES -DNOSIGPIPE
libtool -static -syslibroot ${SDK_ARM} -o ${@.armv7 *.o}
rm *.o
${CC_armv7s} ${CCFLAGS_SO_ARM} -c ${SOURCE_FILES_A} -DMQTT_ASYNC
-DUSE_NAMED_SEMAPHORES -DNOSIGPIPE
libtool -static -syslibroot ${SDK_ARM} -o ${@.armv7s *.o}
rm *.o
${CC_i386} ${CCFLAGS_SO_i386} -c ${SOURCE_FILES_A} -DMQTT_ASYNC
-DUSE_NAMED_SEMAPHORES -DNOSIGPIPE
libtool -static -syslibroot ${SDK_i386} -o ${@.i386 *.o}
rm *.o
lipo -create ${@.armv7} ${@.armv7s} ${@.i386} -output ${@}
${MQTTLIB_DARWIN_S}: ${SOURCE_FILES_S}
-mkdir darwin_x86_64
${CC_armv7} ${CCFLAGS_SO_ARM} -c ${SOURCE_FILES_S} -DOPENSSL
-DUSE_NAMED_SEMAPHORES -DNOSIGPIPE
libtool -static -syslibroot ${SDK_ARM} -L${OPENSSL_DIR}/armv7 -lssl -lcrypto -o ${@.armv7 *.o}
rm *.o
${CC_armv7s} ${CCFLAGS_SO_ARM} -c ${SOURCE_FILES_S} -DOPENSSL
-DUSE_NAMED_SEMAPHORES -DNOSIGPIPE
libtool -static -syslibroot ${SDK_ARM} -L${OPENSSL_DIR}/armv7s -lssl -lcrypto -o ${@.armv7s *.o}
rm *.o
${CC_i386} ${CCFLAGS_SO_i386} -c ${SOURCE_FILES_S} -DOPENSSL -DUSE_NAMED_SEMAPHORES
-DNOSIGPIPE
libtool -static -syslibroot ${SDK_ARM} -L${OPENSSL_DIR}/i386 -lssl -lcrypto -o ${@.i386 *.o}
rm *.o
lipo -create ${@.armv7} ${@.armv7s} ${@.i386} -output ${@}
${MQTTLIB_DARWIN_AS}: ${SOURCE_FILES_AS}
-mkdir darwin_x86_64
${CC_armv7} ${CCFLAGS_SO_ARM} -c ${SOURCE_FILES_AS} -DMQTT_ASYNC -DOPENSSL
-DUSE_NAMED_SEMAPHORES -DNOSIGPIPE
libtool -static -syslibroot ${SDK_ARM} -L${OPENSSL_DIR}/armv7 -lssl -lcrypto -o ${@.armv7 *.o}
rm *.o
${CC_armv7s} ${CCFLAGS_SO_ARM} -c ${SOURCE_FILES_AS} -DMQTT_ASYNC -DOPENSSL
-DUSE_NAMED_SEMAPHORES -DNOSIGPIPE
libtool -static -syslibroot ${SDK_ARM} -L${OPENSSL_DIR}/armv7s -lssl -lcrypto -o ${@.armv7s *.o}
rm *.o
${CC_i386} ${CCFLAGS_SO_i386} -c ${SOURCE_FILES_AS} -DMQTT_ASYNC -DOPENSSL
-DUSE_NAMED_SEMAPHORES -DNOSIGPIPE
libtool -static -syslibroot ${SDK_ARM} -L${OPENSSL_DIR}/i386 -lssl -lcrypto -o ${@.i386 *.o}
rm *.o
lipo -create ${@.armv7} ${@.armv7s} ${@.i386} -output ${@}
.PHONY : clean
clean:
-rm -f *.obj
-rm -f -r darwin_x86_64

```

Download and install the iOS development tools.

For links to client API documentation for the MQTT client libraries, see [MQTT client programming](#)

reference.

You can run an MQTT client for Java app on any platform with JSE 1.5 or above that is "Java Compatible". Log on with a user ID that has administrative authority to IBM MQ.

The server is now ready for you to test your MQTT 3.1 app.

You must have access to an MQTT 3.1 server that supports the MQTT protocol over SSL.

If there is a firewall between your client and the server, check that it does not block MQTT traffic.

You can run an MQTT client for Java app on any platform with JSE 1.5 or above that is "Java Compatible".

See [System Requirements for IBM Mobile Messaging and M2M Client Pack MA9B \(superseded\)](#).

The SSL channels must be started.

Choose an MQTT server to which you can connect the client app.

The statement `rm *.o` deletes all the object files that are created for each library. `lipo` concatenates all three libraries into one file.

Download the [IBM Messaging Telemetry Clients SupportPac](#), or download the clients and libraries under an open source license from the [Eclipse](#)

[Paho](#) project.

Download the IBM Messaging Telemetry Clients SupportPac and install the MQTT SDK.

either IBM MessageSight or IBM MQ as the MQTT server

The server must support the MQTT 3.1 protocol.

The server must support the MQTT 3.1 protocol over SSL.

The MQTT Paho sample applications in the SDK are set up to connect to the Eclipse M2M server by default. See [MQTT Sandbox Server](#). You do not have to configure a server to try the Paho sample applications out.

Create and run the scripts to generate key-pairs and certificates, and configure IBM MQ as the MQTT server.

Check the SSL channels are running and are set up as you expect.

Connections between the MQTT client and the queue manager are always initiated by the MQTT client.

The MQTT client is always the SSL client. Client authentication of the server and server authentication of the MQTT client are both optional.

As an alternative to using SSL, some kinds of Virtual Private Network (VPN), such as IPsec, authenticate the endpoints of a TCP/IP connection. VPN encrypts each IP packet that flows over the network. Once such a VPN connection is established, you have established a trusted network. You can connect MQTT clients to telemetry channels using TCP/IP over the VPN network.

The client always attempts to authenticate the server, unless the client is configured to use a CipherSpec that supports anonymous connection. If the authentication fails, then the connection is not established.

By providing the client with a private signed digital certificate, you can authenticate the MQTT client to WebSphere® MQ. The WebSphere MQ Administrator can force MQTT clients to authenticate themselves to the queue manager using TLS. You can only request client authentication as part of mutual authentication.

Client authentication using TLS relies upon the client having a secret. The secret is the private key of the client in the case of a self-signed certificate, or a key provided by a certificate authority. The key is used to sign the digital certificate of the client. Anyone in possession of the corresponding public key can verify the digital certificate. Certificates can be trusted, or if they are chained, traced back through a certificate chain to a trusted root certificate. Client verification sends all the certificates in the certificate chain provided by the client to the server. The server checks the certificate chain until it finds a certificate it trusts. The trusted certificate is either the public certificate generated from a self-signed certificate, or a root certificate typically issued by a certificate authority. As a final, optional, step the trusted certificate can be compared with a "live" certificate revocation list.

The trusted certificate might be issued by a certificate authority and already included in the JRE certificate store. It might be a self-signed certificate, or any certificate that has been added to the telemetry channel keystore as a trusted certificate.

The telemetry channel has a combined keystore/truststore that holds both the private keys to one or more telemetry channels, and any public certificates needed to authenticate clients. Because an SSL channel must have a keystore, and it is the same file as the channel truststore, the JRE certificate store is never referenced. The implication is that if authentication of a client requires a CA root certificate, you must place the root certificate in the keystore for the channel, even if the CA root certificate is already in the JRE certificate store. The JRE certificate store is never referenced.

Think about the threats that client authentication is intended to counter, and the roles the client and server play in countering the threats. Authenticating the client certificate alone is insufficient to prevent

unauthorized access to a system. If someone else has got hold of the client device, the client device is not necessarily acting with the authority of the certificate holder. Never rely on a single defense against unwanted attacks. At least use a two-factor authentication approach and supplement possession of a certificate with knowledge of private information. For example, use JAAS, and authenticate the client using a password issued by the server.

The primary threat to the client certificate is that it gets into the wrong hands. The certificate is held in a password protected keystore at the client. How does it get placed in the keystore? How does the MQTT client get the password to the keystore? How secure is the password protection? Telemetry devices are often easy to remove, and then can be hacked in private. Must the device hardware be tamper-proof? Distributing and protecting client-side certificates is recognized to be hard; it is called the key-management problem.

A secondary threat is that the device is misused to access servers in unintended ways. For example, if the MQTT application is tampered with, it might be possible to use a weakness in the server configuration using the authenticated client identity.

To authenticate an MQTT client using SSL, configure the telemetry channel, and the client.

The IBM MQ administrator configures telemetry channels at the server. Each channel is configured to accept a TCP/IP connection on a different port number. TLS channels are configured with passphrase protected access to key files. If a TLS channel is defined with no passphrase or key file, the channel does not accept TLS connections.

The client JVM must use the standard socket factory from JSSE. If you are using Java ME, you must ensure that the JSSE package is loaded. If you are using Java SE, JSSE has been included with the JRE since Java version 1.4.1.

The TLS connection requires a number of SSL properties to be set before connecting. You can set the properties either by passing them to the JVM using the `-D` switch, or you can set the properties using the `MqttConnectionOptions.setSSLProperties` method.

If you load a non-standard socket factory, by calling the method `MqttConnectOptions.setSocketFactory(javax.net.SocketFactory)`, then the way SSL settings are passed to the network socket is application defined.

Server authentication using SSL authenticates the server to which you are about to send confidential information to. The client performs the checks matching the certificates sent from the server, against certificates placed in its truststore, or in its JRE cacerts store.

The JRE certificate store is a JKS file, `cacerts`. It is located in `JRE InstallPath\lib\security\`. It is installed with the default password `changeit`. You can either store certificates you trust in the JRE certificate store, or in the client truststore. You cannot use both stores. Use the client truststore if you want to keep the public certificates the client trusts separate from certificates other Java applications use. Use the JRE certificate store if you want to use a common certificate store for all Java applications running on the client. If you decide to use the JRE certificate store review the certificates it contains, to make sure you trust them.

You can modify the JSSE configuration by supplying a different trust provider. You can customize a trust provider to perform different checks on a certificate. In some OGSi environments that have used the MQTT client, the environment provides a different trust provider.

To authenticate the telemetry channel using TLS, configure the server, and the client.

When **SSLCIPH** is used with a telemetry channel, it means TLS Cipher Suite.

The TLS cipher suite is the one supported by the JVM that is running the telemetry (MQXR) service.

If the **SSLCIPH** parameter is blank, no attempt is made to use TLS on the channel.

SSLCIPH specifies the CipherSpec that is used on the channel. The maximum length is 32 characters.

This parameter is valid on all channel types which use transport type **TRPTYPE(TCP)**.

Specify the name of the CipherSpec you are using. The CipherSpecs that can be used with IBM MQ SSL support are shown in the following table. If a specific named CipherSpec is being used, the **SSLCIPH** values at the two ends of a channel must specify the same named CipherSpec.

On IBM MQ for z/OS, you can also specify the two digit hexadecimal code of a CipherSpec, whether or not it appears in the table. On IBM i, you can also specify the two digit hexadecimal code of a CipherSpec, whether or not it appears in the table. Also, on IBM i, installation of AC3 is a prerequisite for the use of SSL.

If you plan to use SHA-2 cipher suites, see [System requirements for using SHA-2 cipher suites with MQTT channels](#).

The value for this parameter is also used to set the value of SECPROT

Download and redistribution of the OpenSSL package is subject to stringent import and export regulation,

and open source licensing conditions. Take careful note of the restrictions and warnings before you decide whether to download the package.

For the C samples, the store is a Privacy-Enhanced Mail (PEM) file. For the Java samples it is a Java keystore (JKS).

Create a script in the client samples directory to compile and run Sample on your chosen platform.

Install a Java development kit (JDK) Version 6 or later.

Because you are developing a Java app for Android, the JDK must come from Oracle. You can get the JDK from [Java SE Downloads](#).

Each successive line that defines the implementation of a target must start with a tab character.

The default file persistence class in the Java SE MQTT client supplied with IBM MQ Telemetry creates a folder with the name: *clientIdIdentifier-tcpHostNameport* or *clientIdIdentifier-sslHostNameport* in the client working directory. The folder name tells you the hostName and port used in the connection attempt

Open the MQTT client sample Java app for Android.

Connect to an MQTT server.

Build output is produced in the current directory.

MQTTCLIENT_DIR must point to the base directory containing the MQTT client source code.

Default MQTTCLIENT_DIR is the current directory

Default OPENSSL_DIR is *sdkroot\openssl*, relative to *sdkroot\sdk\clients\c\mqttv3c\src*

OPENSSL_DIR must point to the base directory containing the OpenSSL build.

Example: `make -f MQTWin.mak MQTTCLIENT_DIR=sdkroot/sdk/clients/c/mqttv3c/src`

Set the build environment, for example:

`%comspec% /k ""C:\Program Files\Microsoft Visual Studio 9.0\VC\vcvarsall.bat"" x86`

`set path=%path%;C:\Program Files\GnuWin32\bin;C:\cygwin\bin`

ifndef MQTTCLIENT_DIR

MQTTCLIENT_DIR = \${CURDIR}

endif

VPATH = \${MQTTCLIENT_DIR}

ifndef OPENSSL_DIR

OPENSSL_DIR = \${MQTTCLIENT_DIR}/../..../openssl-1.0.1c

endif

ALL_SOURCE_FILES = \${wildcard \${MQTTCLIENT_DIR}/*.c}

MQTTLIB = mqttv3c

MQTTDLL = windows_ia32/\${MQTTLIB}.dll

SOURCE_FILES = \${filter-out \${MQTTCLIENT_DIR}/MQTTAsync.c \${MQTTCLIENT_DIR}/SSLSocket.c, \${ALL_SOURCE_FILES}}

MANIFEST = mt -manifest \${MQTTDLL}.manifest -outputresource:\${MQTTDLL}\;2

MQTTLIB_S = mqttv3cs

MQTTDLL_S = windows_ia32/\${MQTTLIB_S}.dll

SOURCE_FILES_S = \${filter-out \${MQTTCLIENT_DIR}/MQTTAsync.c, \${ALL_SOURCE_FILES}}

MANIFEST_S = mt -manifest \${MQTTDLL_S}.manifest -outputresource:\${MQTTDLL_S}\;2

MQTTLIB_A = mqttv3a

MQTTDLL_A = windows_ia32/\${MQTTLIB_A}.dll

SOURCE_FILES_A = \${filter-out \${MQTTCLIENT_DIR}/MQTTClient.c \${MQTTCLIENT_DIR}/SSLSocket.c, \${ALL_SOURCE_FILES}}

MANIFEST_S = mt -manifest \${MQTTDLL_S}.manifest -outputresource:\${MQTTDLL_S}\;2

MQTTLIB_AS = mqttv3as

MQTTDLL_AS = windows_ia32/\${MQTTLIB_AS}.dll

SOURCE_FILES_AS = \${filter-out \${MQTTCLIENT_DIR}/MQTTClient.c, \${ALL_SOURCE_FILES}}

MANIFEST_S = mt -manifest \${MQTTDLL_S}.manifest -outputresource:\${MQTTDLL_S}\;2

CC = cl

CPPFLAGS = /D "WIN32" /D "_UNICODE" /D "UNICODE" /D "_CRT_SECURE_NO_WARNINGS"

CFLAGS = /nologo /c /O2 /W3 /Fd /MD /TC

INC = /I \${MQTTCLIENT_DIR} /I \${MQTTCLIENT_DIR}/..

CPPFLAGS_S = \${CPPFLAGS} /D "OPENSSL"

INC_S = \${INC} /I \${OPENSSL_DIR}/inc32/

LD = link

LINKFLAGS = /nologo /machine:x86 /manifest /dll

```

WINLIBS = kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib\
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib\
odbc32.lib odbccp32.lib ws2_32.lib
IMP = /implib:${@:.dll=.lib}
LIBPDB = /pdb:${@:.dll=.pdb}
LIBMAP = /map:${@:.dll=.map}
WINLIBS_S = ${WINLIBS} crypt32.lib ssleay32.lib libeay32.lib
LIBPATH_S = /LIBPATH:${OPENSSL_DIR}/lib
${MQTTDLL}: ${SOURCE_FILES}
    -mkdir windows_ia32
    -rm ${CURDIR}/MQTTAsync.obj
    ${CC} ${CPPFLAGS} ${CFLAGS} ${INC} /Fo ${SOURCE_FILES}
    ${LD} ${LINKFLAGS} ${IMP} ${LIBPDB} ${LIBMAP} ${WINLIBS} *.obj /out:${MQTTDLL}
    ${MANIFEST}
${MQTTDLL_A}: ${SOURCE_FILES_A}
    -mkdir windows_ia32
    -rm ${CURDIR}/MQTTClient.obj
    ${CC} ${CPPFLAGS} ${CFLAGS} ${INC} /Fo ${SOURCE_FILES_A}
    ${LD} ${LINKFLAGS} ${IMP} ${LIBPDB} ${LIBMAP} ${WINLIBS} *.obj /out:${MQTTDLL_A}
    ${MANIFEST_A}
${MQTTDLL_S}: ${SOURCE_FILES_S}
    -mkdir windows_ia32
    -rm ${CURDIR}/MQTTAsync.obj
    ${CC} ${CPPFLAGS_S} ${CFLAGS} ${INC_S} /Fo ${SOURCE_FILES}
    ${LD} ${LINKFLAGS} ${IMP} ${LIBPDB} ${LIBMAP} ${WINLIBS_S} ${LIBPATH_S} *.obj /out:$
    {MQTTDLL_S}
    ${MANIFEST_S}
${MQTTDLL_AS}: ${SOURCE_FILES_AS}
    -rm ${CURDIR}/MQTTClient.obj
    ${CC} ${CPPFLAGS_S} ${CFLAGS} ${INC_S} /Fo ${SOURCE_FILES_AS}
    ${LD} ${LINKFLAGS} ${IMP} ${LIBPDB} ${LIBMAP} ${WINLIBS_S} ${LIBPATH_S} *.obj /out:$
    {MQTTDLL_AS}
    ${MANIFEST_AS}
all: ${MQTTDLL} ${MQTTDLL_A} ${MQTTDLL_AS} ${MQTTDLL_S}
.PHONY : clean
clean:
    -rm -f *.obj
    -rm -f -r windows_ia32

```

This step is optional on Windows because you can administer IBM MQ as a Windows administrator.

Set the location of the MQTT source code.

Run the makefile in the same directory as the MQTT source files, or set the MQTTCLIENT_DIR command-line parameter:

```
make -f makefile MQTTCLIENT_DIR=sdkroot/SDK/clients/c/mqttv3c/src
```

Add the following lines to the makefile:

The example sets VPATH to the directory where **make** searches for source files that are not explicitly identified; for example all the header files that are required in the build.

Set the location of the OpenSSL libraries.

This step is required to build the SSL versions of the MQTT client for C libraries.

Set the default path to the OpenSSL libraries to same directory as you expanded the MQTT SDK.

Otherwise, set OPENSSL_DIR as a command-line parameter.

OpenSSL is the OpenSSL directory that contains all the OpenSSL subdirectories. You might have to move the directory tree from where you expanded it, because it contains unnecessary empty parent directories. Select all the source files that are required to build each MQTT library. Also, set the name and location of the MQTT library to build.

The source files depend on whether you are building a synchronous or asynchronous library, and whether the library includes SSL or not.

Add the following line to the makefile to list all the MQTT source files:

Set the keystore location and attributes with IBM MQ Explorer, or with the **DEFINE CHANNEL** command; see [DEFINE CHANNEL \(MQTT\)](#). Multiple channels can share a keystore.

MQTT non-phrases

```
MQTTClient mqttClient = new MqttClient( "ssl://www.example.org:8884", "clientId1");
mqttClient.connect();
```

SampleAsyncCallback

SampleAsyncCallback is in the `org.eclipse.paho.client.mqttv3` package. It calls the asynchronous MQTT API. The asynchronous API does not wait for MQTT to complete processing a call; it returns to the application. The application carries on with other tasks, then waits for the next event to arrive for it to process. MQTT posts an event notification back to the application when it completes processing. The event driven MQTT interface is suited to the service and activity programming model of Android and other event driven operating systems.

As an example, look at how the `mqttExerciser` sample integrates MQTT into Android using the service and activity programming model.

SampleAsyncWait

SampleAsyncWait is in the `org.eclipse.paho.client.mqttv3` package. It uses the asynchronous MQTT API; it waits on a different thread until an action completes. The main thread can do other work until it synchronizes on the thread that is waiting for the MQTT action to complete.

The example command files create the certificates and certificate stores as described in the steps in the task. In addition, the example sets up the MQTT client queue manager to use the server certificate store. The example deletes and re-creates the queue manager by calling the `SampleMQM.bat` script that is provided with IBM MQ.

initcert.bat

`initcert.bat` sets the names and paths to certificates and other parameters that are required by the **keytool** and **openssl** commands. The settings are described in comments in the script.

```
@echo off
@rem Set the path where you installed the MQTT SDK
@rem and short cuts to the samples directories.
set SDKRoot=C:\MQTT
set jsamppath=%SDKRoot%\sdk\clients\java\samples
set csamppath=%SDKRoot%\sdk\clients\c\samples
```

```
@rem Set the paths to Version 7 of the JDK
@rem and to the directory where you built the openssl package.
@rem Set short cuts to the tools.
set javapath=C:\Program Files\IBM\Java70
set keytool="%javapath%\jre\bin\keytool.exe"
set ikeyman="%javapath%\jre\bin\ikeyman.exe"
set openssl=%SDKRoot%\openssl
set runopenssl="%openssl%\bin\openssl"
```

```
@rem Set the path to where certificates are to be stored,
@rem and set global security parameters.
@rem Omit set password, and the security tools prompt you for passwords.
@rem Validity is the expiry time of the certificates in days.
set certpath=%SDKRoot%\Certificates
set password=password
set validity=5000
set algorithm=RSA
```

```
@rem Set the certificate authority (CA) jks keystore and certificate parameters.
@rem Omit this step, unless you are defining your own certificate authority.
@rem The CA keystore contains the key-pair for your own certificate authority.
@rem You must protect the CA keystore.
@rem The CA certificate is the self-signed certificate authority public certificate.
@rem It is commonly known as the CA root certificate.
set caalias=caalias
set cadname="CN=mqttca.ibm.id.com, OU=ID, O=IBM, L=Hursley, S=Hants, C=GB"
set cakeypass=%password%
@rem ca keystore
set cajkskeystore=%certpath%\cakeystore.jks
set cajkskeystorepass=%password%
@rem ca certificate (root certificate)
set cacert=%certpath%\cacert.cer
```

```
@rem Set the server jks keystore and certificate parameters.
@rem The server keystore contains the key-pair for the server.
@rem You must protect the server keystore.
@rem If you then export the server certificate it is self-signed.
@rem Alternatively, if you export a certificate signing request (CSR)
@rem from the server keystore for the server key,
@rem and import the signed certificate back into the same keystore,
@rem it forms a certificate chain.
@rem The certificate chain links the server certificate to the CA.
@rem When you now export the server certificate,
@rem the exported certificate includes the certificate chain.
set srvalias=srvalias
set srvidname="CN=mqttserver.ibm.id.com, OU=ID, O=IBM, L=Hursley, S=Hants, C=GB"
set srvkeypass=%password%
@rem server keystores
set srvjkskeystore=%certpath%\srvkeystore.jks
set srvjkskeystorepass=%password%
@rem server certificates
set srvcertreq=%certpath%\srvcertreq.csr
set srvcertcasigned=%certpath%\srvcertcasigned.cer
set srvcertselfsigned=%certpath%\srvcertselfsigned.cer
```

```
@rem Set the client jks keystore and certificate parameters
@rem Omit this step, unless you are authenticating clients.
@rem The client keystore contains the key-pair for the client.
@rem You must protect the client keystore.
@rem If you then export the client certificate it is self-signed.
@rem Alternatively, if you export a certificate signing request (CSR)
@rem from the client keystore for the client key,
@rem and import the signed certificate back into the same keystore,
@rem it forms a certificate chain.
@rem The certificate chain links the client certificate to the CA.
@rem When you now export the client certificate,
@rem the exported certificate includes the certificate chain.
set cltalias=cltalias
set cltdname="CN=mqttclient.ibm.id.com, OU=ID, O=IBM, L=Hursley, S=Hants, C=GB"
set cltkeypass=%password%
@rem client keystores
set cltjkskeystore=%certpath%\cltkeystore.jks
set cltjkskeystorepass=%password%
set cltcertreq=%certpath%\cltcertreq.csr
set cltcertcasigned=%certpath%\cltcacertsigned.cer
set cltcertselfsigned=%certpath%\cltcertselfsigned.cer
```

```
@rem Set the paths to the client truststores signed by CA and signed by server key.
@rem You only need to define one of the truststores.
@rem A truststore holds certificates that you trust,
@rem which are used to authenticate untrusted certificates.
@rem In this example, when the client authenticates the MQTT server it connects to,
@rem it authenticates the certificate it is sent by the server
@rem with the certificates in its truststore.
@rem For example, the MQTT server sends its server certificate,
@rem and the client authenticates it with either the same server certificate
@rem that you have stored in the cltsrvtruststore.jks truststore.
```

cleancert.bat

The commands in the cleancert.bat script delete the MQTT client queue manager to ensure that the server certificate store is not locked, and then delete all the keystores and certificates that are created by the sample security scripts.

```
@rem Delete the MQTT sample queue manager, MQXR_SAMPLE_QM
call "%MQ_FILE_PATH%\bin\setmqenv" -s
endmqm -i %qm%
dlmqm %qm%
```

```
@rem Erase all the certificates and keystores created by the sample scripts.
erase %cajkskeystore%
erase %cacert%
erase %srvjkskeystore%
erase %srvcertreq%
erase %srvcertcasigned%
erase %srvcertselfsigned%
erase %cltjkskeystore%
erase %cltp12keystore%
erase %cltpemkeystore%
erase %cltcertreq%
erase %cltcertcasigned%
erase %cltcertselfsigned%
erase %cltcajkstruststore%
erase %cltcap12truststore%
erase %cltcapemtruststore%
erase %cltsrvjkskeystore%
erase %cltsrvp12truststore%
erase %cltsrvpemtruststore%
erase %mqlog%
@echo Cleared all certificates
dir %certpath%\*.* /b
```

Figure 2. cleancert.bat

genkeys.bat

The commands in the genkeys.bat script create key-pairs for your private certificate authority, the server, and a client.

```
@rem
@echo
@echo Generate %caalias%, %srvalias%, and %cltalias% key-pairs in %cajkskeystore%,
%srvjkskeystore%, and %cltjkskeystore%
@rem
@rem -- Generate a client certificate and a private key pair
@rem Omit this step, unless you are authenticating clients.
%keytool% -genkeypair -noprompt -alias %cltalias% -dname %cltdname% -keystore
%cltjkskeystore% -storepass %cltjkskeystorepass% -keypass %cltkeypass% -keyalg %algorithm%
-validity %validity%

@rem -- Generate a server certificate and private key pair
%keytool% -genkeypair -noprompt -alias %srvalias% -dname %srvdname% -keystore
%srvjkskeystore% -storepass %srvjkskeystorepass% -keypass %srvkeypass% -keyalg %algorithm%
-validity %validity%

@rem Create CA, client and server key-pairs
@rem -- Generate a CA certificate and private key pair - The extension asserts this is a
certificate authority certificate, which is required to import into firefox
%keytool% -genkeypair -noprompt -ext bc=ca:true -alias %caalias% -dname %cadname%
-keystore %cajkskeystore% -storepass %cajkskeystorepass% -keypass %cakeypass% -keyalg
%algorithm% -validity %validity%
```

Figure 3. genkeys.bat

sscerts.bat

The commands in the sscerts.bat script export the client and server self-signed certificates from their keystores, and import the server certificate into the client truststore, and the client certificate

into the server keystore. The server does not have a truststore. The commands create a client truststore in PEM format from the client JKS truststore.

```
@rem
@echo
@echo Export self-signed certificates: %srcertselfsigned% and %cltcertselfsigned%
@rem Export Server public certificate
%keytool% -exportcert -noprompt -rfc -alias %srvalias% -keystore %srvjkskeystore%
-storepass %srvjkskeystorepass% -file %srcertselfsigned%
@rem Export Client public certificate
@rem Omit this step, unless you are authenticating clients.
%keytool% -exportcert -noprompt -rfc -alias %cltalias% -keystore %cltjkskeystore%
-storepass %cltjkskeystorepass% -file %cltcertselfsigned%
```

```
@rem
@echo
@echo Add selfsigned server certificate %srcertselfsigned% to client truststore:
%cltsrvjkstruststore%
@rem Import the server certificate into the client-server truststore (for server self-
signed authentication)
%keytool% -import -noprompt -alias %srvalias% -file %srcertselfsigned% -keystore
%cltsrvjkstruststore% -storepass %cltsrvjkstruststorepass%
```

```
@rem
@echo
@echo Add selfsigned client certificate %cltcertselfsigned% to server truststore:
%srvjkskeystore%
@rem Import the client certificate into the server truststore (for client self-signed
authentication)
@rem Omit this step, unless you are authenticating clients.
%keytool% -import -noprompt -alias %cltalias% -file %cltcertselfsigned% -keystore
%srvjkskeystore% -storepass %srvjkskeystorepass%
```

```
@rem
@echo
@echo Create a pem client-server truststore from the jks client-server truststore:
%cltsrvpemtruststore%
@rem
%keytool% -importkeystore -noprompt -srckeystore %cltsrvjkstruststore% -destkeystore
%cltsrvp12truststore% -srcstoretype jks -deststoretype pkcs12 -srcstorepass
%cltsrvjkstruststorepass% -deststorepass %cltsrvp12truststorepass%
%openssl%\bin\openssl pkcs12 -in %cltsrvp12truststore% -out %cltsrvpemtruststore% -passin
pass:%cltsrvp12truststorepass% -passout pass:%cltsrvpemtruststorepass%@rem
@rem
@echo
@echo Create a pem client keystore from the jks client keystore
@rem Omit this step, unless you are configuring a C or iOS client.
@rem
%keytool% -importkeystore -noprompt -srckeystore %cltjkskeystore% -destkeystore
%cltp12keystore% -srcstoretype jks -deststoretype pkcs12 -srcstorepass
%cltjkskeystorepass% -deststorepass %cltp12keystorepass%
%openssl%\bin\openssl pkcs12 -in %cltp12keystore% -out %cltpemkeystore% -passin
pass:%cltp12keystorepass% -passout pass:%cltpemkeystorepass%
```

Figure 4. *sscerts.bat*

cacerts.bat

The script imports the certificate authority root certificate into the private keystores. The CA root certificate is needed to create the keychain between the root certificate and the signed certificate. The *cacerts.bat* script exports the client and server certificate requests from their keystores. The script signs the certificate requests with the key of the private certificate authority in the *cajkskeystore.jks* keystore, then imports the signed certificates back into the same keystores from which the requests came. The import creates the certificate chain with the CA root certificate. The script creates a client truststore in PEM format from the client JKS truststore.

```

@rem
@echo
@echo Export self-signed certificates: %cacert%
@rem
@rem Export CA public certificate
%keytool% -exportcert -noprompt -rfc -alias %caalias% -keystore %cajkskeystore% -storepass
%cajkskeystorepass% -file %cacert%

```

```

@rem
@echo
@echo Add CA to server key and client key and truststores: %srvjkskeystore%,
%cltjkskeystore%, %cltcajkstruststore%,
@rem The CA certificate is necessary to create key chains in the client and server
keystores,
@rem and to certify key chains in the server keystore and the client truststore
@rem
@rem Import the CA root certificate into the server keystore
%keytool% -import -noprompt -alias %caalias% -file %cacert% -keystore %srvjkskeystore%
-storepass %srvjkskeystorepass%
@rem Import the CA root certificate into the client keystore
@rem Omit this step, unless you are authenticating clients.
%keytool% -import -noprompt -alias %caalias% -file %cacert% -keystore %cltjkskeystore%
-storepass %cltjkskeystorepass%
@rem Import the CA root certificate into the client ca-truststore (for ca chained
authentication)
%keytool% -import -noprompt -alias %caalias% -file %cacert% -keystore %cltcajkstruststore%
-storepass %cltcajkstruststorepass%

```

```

@rem
@echo
@echo Create certificate signing requests: %srvcertreq% and %cltcertreq%
@rem
@rem Create a certificate signing request (CSR) for the server key
%keytool% -certreq -alias %srvalias% -file %srvcertreq% -keypass %srvkeypass% -keystore
%srvjkskeystore% -storepass %srvjkskeystorepass%
@rem Create a certificate signing request (CSR) for the client key
%keytool% -certreq -alias %cltalias% -file %cltcertreq% -keypass %cltkeypass% -keystore
%cltjkskeystore% -storepass %cltjkskeystorepass%

```

```

@rem
@echo
@echo Sign certificate requests: %srvcertcasigned% and %cltcertcasigned%
@rem The requests are signed with the ca key in the cajkskeystore.jks keystore
@rem
@rem Sign server certificate request
%keytool% -gencert -infile %srvcertreq% -outfile %srvcertcasigned% -alias %caalias%
-keystore %cajkskeystore% -storepass %cajkskeystorepass% -keypass %cakeypass%
@rem Sign client certificate request
@rem Omit this step, unless you are authenticating clients.
%keytool% -gencert -infile %cltcertreq% -outfile %cltcertcasigned% -alias %caalias%
-keystore %cajkskeystore% -storepass %cajkskeystorepass% -keypass %cakeypass%

```

```

@rem
@echo
@echo Import the signed certificates back into the keystores to create the key chain:
%srvjkskeystore% and %cltjkskeystore%
@rem
@rem Import the signed server certificate
%keytool% -import -noprompt -alias %srvalias% -file %srvcertcasigned% -keypass %srvkeypass%
-keystore %srvjkskeystore% -storepass %srvjkskeystorepass%
@rem Import the signed client certificate and key chain back into the client keystore
%keytool% -import -noprompt -alias %cltalias% -file %cltcertcasigned% -keypass %cltkeypass%
-keystore %cltjkskeystore% -storepass %cltjkskeystorepass%
@rem
@rem The CA certificate is needed in the server keystore, and the client truststore
@rem to verify the key chain sent from the client or server
@echo Delete the CA certificate from %cltjkskeystore%: it causes a problem in converting
keystore to pem
@rem Omit this step, unless you are authenticating clients.
%keytool% -delete -alias %caalias% -keystore %cltjkskeystore% -storepass
%cltjkskeystorepass%

```

```

@rem
@echo
@echo Create a pem client-ca truststore from the jks client-ca truststore:
%cltcapemtruststore%

```


mqcerts.bat

The script lists the keystores and certificates in the certificate directory. It then creates the MQTT sample queue manager and configures the secure telemetry channels.

```
@echo
@echo -----
@echo List keystores and certificates
dir %certpath%\*.* /b

@rem
@echo Create queue manager and define mqtt channels and certificate stores
call "%MQ_FILE_PATH%\mqxr\Samples\SampleMQM" >> %mqlog%
echo DEFINE CHANNEL(%chlreq%) CHLTYPE(MQTT) TRPTYPE(TCP) PORT(%portsslreq%)
SSLCAUTH(%authreq%) SSLKEYR('%srvjkskeystore%') SSLKEYP('%srvjkskeystorepass%')
MCAUSER(%mcauser%) | runmqsc %qm% >> %mqlog%
echo DEFINE CHANNEL(%chlopt%) CHLTYPE(MQTT) TRPTYPE(TCP) PORT(%portsslopt%)
SSLCAUTH(%authopt%) SSLKEYR('%srvjkskeystore%') SSLKEYP('%srvjkskeystorepass%')
MCAUSER(%mcauser%) | runmqsc %qm% >> %mqlog%
echo DEFINE CHANNEL(%chlsslreqws%) CHLTYPE(MQTT) TRPTYPE(TCP) PORT(%portsslreqws%)
SSLCAUTH(%authreq%) SSLKEYR('%srvjkskeystore%') SSLKEYP('%srvjkskeystorepass%')
MCAUSER(%mcauser%) | runmqsc %qm% >> %mqlog%
echo DEFINE CHANNEL(%chlssloptws%) CHLTYPE(MQTT) TRPTYPE(TCP) PORT(%portssloptws%)
SSLCAUTH(%authopt%) SSLKEYR('%srvjkskeystore%') SSLKEYP('%srvjkskeystorepass%')
MCAUSER(%mcauser%) | runmqsc %qm% >> %mqlog%
echo DEFINE CHANNEL(%chlws%) CHLTYPE(MQTT) TRPTYPE(TCP) PORT(%portws%)
MCAUSER(%mcauser%) | runmqsc %qm% >> %mqlog%
@echo MQ logs saved in %mqlog%
```

Figure 6. mqcerts.bat

Get a copy of the IBM MQ installation materials and a license in one of the following ways:

1. Ask your IBM MQ administrator for the installation materials, and to confirm you can accept the license agreement.
2. Get a 90-day evaluation copy of IBM MQ. See [Evaluate: IBM MQ](#).
3. Buy IBM WebSphere MQ. See [IBM MQ product page](#).

Secure the SSL channel with either certificate authority signed keys, or self-signed keys.

There is no installation program, you just expand the downloaded file.

1. Download the [IBM Messaging Telemetry Clients SupportPac](#), or download the clients and libraries under an open source license from the [Eclipse Paho project](#).
2. Create a folder where you are going to install the SDK.

You might want to name the folder MQTT. The path to this folder is referred to here as *sdkroot*.

3. Expand the compressed IBM Messaging Telemetry Clients SupportPac file contents into *sdkroot*. The expansion creates a directory tree that starts at *sdkroot\SDK*.

There is a similar limitation for the MQTT client for Java. If the client code is running on a Java 1.6 JRE from IBM, the required SHA-2 cipher suites must be explicitly enabled. In order to use these suites, the client must also set the SSL context to a value that supports Version 1.2 of the Transport Layer Security (TLS) protocol. For example:

```
MqttConnectOptions mqttConnectOptions = new MqttConnectOptions();
java.util.Properties sslClientProps = new java.util.Properties();
sslClientProps.setProperty("com.ibm.ssl.keyStore", sslKeys.clientKeyStore);
sslClientProps.setProperty("com.ibm.ssl.keyStorePassword", sslKeys.clientStorePassword);
sslClientProps.setProperty("com.ibm.ssl.trustStore", sslKeys.clientKeyStore);
sslClientProps.setProperty("com.ibm.ssl.trustStorePassword", sslKeys.clientStorePassword);
sslClientProps.setProperty("com.ibm.ssl.protocol", "TLSv1.2");
sslClientProps.setProperty("com.ibm.ssl.enabledCipherSuites",
    "SSL_RSA_WITH_AES_256_CBC_SHA256" );
mqttConnectOptions.setSSLProperties(sslClientProps);
```

On IBM MQ, type the following command into a command window:

- ▶ **Linux**

```
echo 'DISPLAY CHSTATUS(SSL*) CHLTYPE(MQTT) ALL' | runmqsc MQXR_SAMPLE_QM
echo 'DISPLAY CHANNEL(SSL*) CHLTYPE(MQTT) ALL' | runmqsc MQXR_SAMPLE_QM
```

- ▶ **Windows**

```
echo DISPLAY CHSTATUS(SSL*) CHLTYPE(MQTT) ALL | runmqsc MQXR_SAMPLE_QM
echo DISPLAY CHANNEL(SSL*) CHLTYPE(MQTT) ALL | runmqsc MQXR_SAMPLE_QM
```

When you request a personal certificate, you specify a key size for the public and private key pair. The key size that is used during the SSL handshake can depend on the size stored in the certificate and on the CipherSpec:

- ▶ **z/OS** ▶ **ULW** On z/OS, UNIX, Linux, and Windows, when a CipherSpec name includes `_EXPORT`, the maximum handshake key size is 512 bits. If either of the certificates exchanged during the SSL handshake has a key size greater than 512 bits, a temporary 512-bit key is generated for use during the handshake.
- ▶ **ULW** On UNIX, Linux, and Windows, when a CipherSpec name includes `_EXPORT1024`, the handshake key size is 1024 bits.
- Otherwise the handshake key size is the size stored in the certificate.

For further information, see the:

- Eclipse Paho web site, for the latest telemetry clients and an Android sample.
- MA9C: [IBM Messaging Telemetry Clients SupportPac](#), which includes a similar set of clients to Paho, plus samples for Java and C.
- Deprecated MA9B: [Mobile Messaging and M2M Client Pack](#) includes a sample for iOS that was written in the IBM WebSphere MQ 7.5 time frame.

IBM MQ Explorer Requirements

Before you install IBM MQ Explorer, check the minimum requirements that are listed in this topic.

IBM MQ Explorer is available for Windows and Linux and can be installed either as part of the product installation, or from the stand-alone IBM MQ Explorer support pack MS0T.

- The product version is available for Windows x86_64 and Linux x86_64.
- The support pack version is available for Windows x86_64, and Linux x86_64.

The requirements to install IBM MQ Explorer either as part of the product installation, or from the stand-alone IBM MQ Explorer support pack MS0T, include:

- 512 MB RAM
- 1 GHz processor
- At least 300 MB available disk space
- A suitable monitor for the operating system with a screen size of at least 1024x768
- ▶ **V 9.0.4** On Linux, GTK2 including the GTK2-engines, which contain the GTK2 themes. The minimum GTK2 level depends on the version of IBM MQ:
 - On IBM MQ 9.0 Long Term Support, or IBM MQ 9.0 Continuous Delivery releases, prior to IBM MQ 9.0.4, GTK+ version 2.10.0 or later is supported.
 - From IBM MQ 9.0.4, GTK+ version 2.18.0, or later, is supported.
- Bitstream-vera-fonts (applies to Linux only)

Related information

[Windows 7 system requirements](#)

[Windows 8 system requirements](#)

What's new and what's changed in IBM MQ Explorer

Learn about the main new and changed functions in IBM MQ Explorer.

New features in IBM MQ Explorer 9.0

IBM MQ Explorer is now built on Eclipse 4.4.2 instead of Eclipse 4.3.2.

For more information see [“Installing into Eclipse environments”](#) on page 24.

New features in IBM MQ Explorer 9.0.4

V 9.0.4

IBM MQ Explorer is now built on Eclipse 4.6.3 instead of Eclipse 4.4.2.

For more information see [“Installing into Eclipse environments”](#) on page 24.

New features in IBM MQ Explorer 8.0

IBM MQ Explorer is now built on Eclipse 4.3.1 instead of Eclipse 3.6.2.

For more information, see [“Installing into Eclipse environments”](#) on page 24.

Support for topic host routing for publish/subscribe clusters

In previous versions, when you configure a clustered topic on a queue manager, all queue managers in the cluster become aware of all other queue managers in the cluster. When performing publish and subscribe operations, each queue manager then connects directly to all the others. This approach is still available in IBM MQ 8.0, where it is known as *direct routing*.

In IBM MQ 8.0, an alternative approach has also been added, known as *topic host routing*. With this approach, all queue managers in the cluster become aware of the cluster queue managers that host the routed topic definitions. When performing publish and subscribe operations, queue managers in the cluster connect only to these topic host queue managers, and not directly to each other. The topic host queue managers are responsible for routing publications from queue managers on which publications are published to queue managers with matching subscriptions.

To support topic host routing, the following parameters are added:

- **Cluster publication route.** The routing behavior of publications between queue managers in a cluster. This is set on the cluster tab of a topic object and is displayed on the cluster tab of a topic object and when displaying cluster topics.
- **Cluster object state.** The current state of the clustered topic definition. This is displayed on the cluster tab of a topic object, and when displaying cluster topics.
- **Version.** The version of the IBM MQ installation that the cluster queue manager is associated with. This is displayed on the cluster-sender channels tab of the queue manager clusters display.

Support to better understand the size of your system

The following parameters are added to the publish/subscribe information reported. They are displayed on the Publish/Subscribe status page for a given queue manager.

- **Sub count.** Shows the total number of subscriptions against the local topic tree.
- **Topic count.** Shows the total number of topic nodes in the local topic tree.

For more information, see [“Queue manager Publish/Subscribe Engine status attributes”](#) on page 612.

New connection details properties

For more information, see [“Connection details properties”](#) on page 558.

CHKLOCL

Setting CHCKLOCL to Required for administrators or Required for all stops you from locally administering the queue manager by way of the **runmqsc** commands unless you specify the -u UserID parameter on the **runmqsc** command line.

For more information, see the CHCKLOCL MQSC parameter explanation in the [“User ID + Password page”](#) on page 498 section of [“Authentication information properties”](#) on page 494.

Security enabled remote queue manager connections

The SSL cipher specification RC2_MD5_EXPORT is no longer supported. Connections that use this cipher specification and that are imported into IBM MQ Explorer 8.0 have a blank SSL cipher specification setting. A new cipher specification must be selected.

If a connection that was using this cipher specification is imported into IBM MQ Explorer 8.0 and then used without changing it, a dialog that contains the IBM MQ error message AMQ4199 is displayed.

Deprecation of specific SSLv3 Cipher Suites

The three SSL cipher specifications listed in [Java and JMS: changes to CipherSuite support](#) in the IBM MQ 8.0 online product documentation are no longer supported.

However, you can re-enable other SSLv3 ciphers. See [Deprecation: SSLv3 Ciphers](#) in the IBM MQ 8.0 online product documentation.



New for z/OS

For z/OS, the following changes are described in [“Queue manager properties”](#) on page 349 and [“Channel properties”](#) on page 426:

- New [“Statistics monitoring \(z/OS\)”](#) on page 403 section.
- Link added from [“Statistics monitoring \(Multiplatforms\)”](#) on page 377 to [“Statistics monitoring \(z/OS\)”](#) on page 403.
- New **Security policies**, ***Maximum ACE pool size (KB)**, and **Excluded operator messages** parameters added to [“System \(z/OS\)”](#) on page 408.
- **Channel statistics** parameter description enhanced in [“Statistics page”](#) on page 454.
- Clarification added to the **Cluster channel names** parameter for z/OS. For more information, see the **Cluster channel names** attribute in the [“Extended page”](#) on page 415 of [“IBM MQ queue properties”](#) on page 412.

New features in IBM WebSphere MQ Explorer 7.5

Integration of Managed File Transfer and Advanced Message Security

A key new feature of IBM WebSphere MQ 7.5 is the integration of Managed File Transfer and Advanced Message Security function, the use of which is subject to appropriate licensing entitlements. There is a single installation where you choose the required components.

The Managed File Transfer and Advanced Message Security extensions are always present in IBM MQ Explorer, both in the server component and the stand-alone version from SupportPac MS0T, and become active as required. You now always see a Managed File Transfer node in the IBM MQ Explorer **Navigator** view.

If you have an existing managed file transfer configuration on the computer, created from the earlier separate product, it automatically opens under the Managed File Transfer node. Alternatively, right-clicking on the node opens a new configuration wizard to select your coordination queue manager and command queue manager. This feature is useful if you are using IBM MQ Explorer to connect to remote queue managers that are configured for use with Managed File Transfer. When you set up your configuration, you can then use IBM MQ Explorer to do a number of operations, including scheduling transfers and monitoring their progress.

Advanced Message Security provides a high level of protection for sensitive data that is flowing through your IBM WebSphere MQ network. Messages can be both encrypted and also signed so that the identity of the sender can be confirmed. When IBM MQ Explorer connects to a queue manager,

if it detects the Advanced Message Security capability on the queue manager, it adds a Security Policies node under the queue manager in the navigator view. From there, you can create security policies to protect specific queues.

Improved multi-version support

Another key new feature of this release of IBM MQ Explorer is improved multi-version support. From IBM WebSphere MQ 7.1, you are able to install multiple copies of IBM WebSphere MQ on the same computer. IBM MQ Explorer has improved function to further simplify working with multiple installations of IBM WebSphere MQ.

The **Welcome** screen now has a new link to **View installations**. Clicking this link displays all of your IBM WebSphere MQ installations. Each installation might have its own copy of IBM MQ Explorer, so the installation that is running IBM MQ Explorer is shown first. All other installations are then shown in the second table. Each installation has a name, version, and installation path, and optionally a description. One of the installations might be set as the primary one on the computer (setting as primary sets various system-wide settings such as environment variables on Windows systems). The **State** is Available unless there is an issue with the installation, and the **Identifier** is allocated at installation time and is for internal use.

There is a new **Transfer queue managers to this installation** function on the dialog. This control opens a new wizard, which can also be opened by right-clicking on the Queue Managers node in the navigator tree and selecting **Transfer Queue Managers....** This new wizard enables you to transfer one or more queue managers from other installations to the current installation. This wizard is equivalent to the **setmqm** command, but saves having to type the required paths and parameters. Only stopped queue managers can be transferred; running queue managers are shown for reference. Queue managers can be pulled only into the current installation; they cannot be pushed out into other installations. After you click Transfer, the **setmqm** command is invoked with the selected queue managers. If successful, the navigator tree updates to include the transferred queue managers. If there are any problems, a dialog is shown with the error message from the command. You can also use drag and drop, and IBM MQ Explorer import and export function, to transfer queue managers. For drag and drop, you must run two instances of IBM MQ Explorer at the same time, from different installations. You can then drag a stopped queue manager out from one installation and drop it into the queue managers folders of the other. You then see a confirmation dialog.

If you export IBM MQ Explorer settings to a file, under the **Connection Information** category there is now an option to save details for local queue managers. If you then import that file into IBM MQ Explorer from a different installation, you get the same prompt as you do if you want to transfer the queue manager. You can drag the exported XML file into IBM MQ Explorer to initiate the import.

Reduced overhead

To reduce the overhead of maintaining the Navigator view, the amount of information about the state of the Navigator view of IBM MQ Explorer persistent on disk is reduced. As a result of this change, the expansion state of tree nodes in the Navigator view no longer persists when IBM MQ Explorer is restarted.

New features in IBM WebSphere MQ Explorer 7.1

The following table lists the additional tasks that you can perform in IBM MQ Explorer in IBM WebSphere MQ 7.1.

IBM MQ Explorer is repackaged in IBM WebSphere MQ 7.1 to be self-contained, no longer requiring the installation of a separate package for Eclipse technology. For more details about the IBM WebSphere Eclipse platform no longer being packaged with IBM MQ Explorer, see [IBM MQ Explorer changes](#) in the IBM WebSphere MQ online product documentation.

The multi-version capabilities of IBM WebSphere MQ 7.1 enable you to install a copy of IBM MQ Explorer as part of each installation of IBM WebSphere MQ and run them independently. You can view the installation properties of local and remote queue managers.

Performance improvements have been made to IBM WebSphere MQ Explorer 7.1, across various areas. Startup time is reduced and most operations that involve large numbers of queue managers are now

quicker, including connecting to remote queue managers by using multiple selection, and managing the list of queue managers in the Navigator view. The time that is taken to populate tables with many objects such as queues is also reduced.

Note: Some of the links in this topic work only when clicked from inside the installed IBM MQ Explorer. If you are reading this material online, then some of the links might not work.

New features in IBM WebSphere MQ Explorer 7.1	More information
Support for multiple installations.	See “Multiple installations of IBM MQ Explorer” on page 23.
Installing into Eclipse environments.	See “Installing into Eclipse environments” on page 24.

New features since IBM WebSphere MQ 6.x	More information
Create and configure IBM WebSphere MQ service definitions by using IBM MQ Explorer.	See “Creating and configuring a service definition” on page 228.
Create and configure IBM WebSphere MQ administered objects by using IBM MQ Explorer.	See “Creating and configuring JMS administered objects” on page 204.
Manage IBM WebSphere MQ object authorities by using an authorization service in IBM MQ Explorer.	See “Managing object authorities with an authorization service” on page 165.
Test your object definitions for problems or write tests to enforce rules in object definitions.	See “Testing your object definitions for problems” on page 60.
Configure a IBM WebSphere MQ publish/subscribe engine in IBM MQ Explorer.	See “Queue manager properties” on page 349.
Get context-sensitive help from the pop-up help throughout IBM MQ Explorer.	In a dialog, or over a tree item, press F1 on Windows or Ctrl+F1 on Linux (x86 and x86-64 platforms) to display the pop-up help. The pop-up help contains a brief description and some links to take you to more information in the IBM MQ Explorer help. Dialogs now include a small help icon to display the pop-up help when clicked, as well as pressing F1 or Ctrl-F1. See: “Displaying context-sensitive help (pop-up window help)” on page 25
Create and configure IBM WebSphere MQ classes for JMS administered objects by using IBM MQ Explorer.	See “Creating and configuring JMS administered objects” on page 204.
Manage IBM WebSphere MQ object authorities by using an authorization service in IBM MQ Explorer.	See “Managing object authorities with an authorization service” on page 165.
Group queue managers in folders, and make actions to all the queue managers in the set. Queue managers that belong to particular applications, departments, or companies can be grouped.	See “Creating and configuring a queue manager set” on page 233.
Tightly integrated support for Publish and Subscribe messaging that simplifies the development, configuration, and deployment of event-driven messaging.	See “Configuring publish/subscribe messaging” on page 117.
Optimizations to the JMS layer, tightly integrating JMS services into IBM WebSphere MQ queue manager internals. Consolidated administration and security for Publish and Subscribe, JMS, and XMS messaging into the standard IBM WebSphere MQ model.	See “Creating and configuring JMS administered objects” on page 204.

New features since IBM WebSphere MQ 6.x	More information
Welcome pages are a quick and easy way to find links for tutorials, applications, and education.	See “Introduction to IBM MQ Explorer” on page 5.

Related reference

“Icons in IBM MQ Explorer” on page 310

IBM MQ Explorer uses icons to represent the different objects, such as queue managers, queues, and channels.

Launching IBM MQ Explorer

You can launch IBM MQ Explorer by using one of three methods; by using the system menu, the **MQExplorer** command, or the **strmqcfg** command.

To launch IBM MQ Explorer by using the system menu on Linux, or the start menu on Windows, you must left-click on the installation that you want to launch. On Windows, open the start menu, and select the IBM MQ Explorer installation entry under the **IBM MQ** group that corresponds to the installation that you want to launch. Each instance of IBM MQ Explorer listed is identified by the name that you chose for its installation.

On Linux, the system menu entry for IBM MQ Explorer is added to the **Development** category; where it appears within the system menu is dependent on your Linux distribution (SUSE or Red Hat), and your desktop environment (GNOME or KDE). On SUSE, left-click **Computer > More Applications...**, and find the installation of IBM MQ Explorer that you want to launch under the **Development** category. On Red Hat, the installation of IBM MQ Explorer that you want to launch can be found under **Applications > Programming**.

The **MQExplorer** command and the **strmqcfg** command are stored in *MQ_INSTALLATION_PATH/bin*.

MQExplorer.exe (the **MQExplorer** command) supports standard Eclipse runtime options including the following:

-clean

Clean the caches used by the eclipse runtime to store bundle dependency resolution and eclipse extension registry data. Using this option forces eclipse to reinitialize these caches.

-initialize

Initializes the configuration being run. All runtime-related data structures and caches are refreshed. Any user/plugin defined configuration data is not purged. No application is run, any product specifications are ignored and no UI is presented (for example, the splash screen is not drawn).

You can enter either the **MQExplorer** command or the **strmqcfg** command from the command line to launch IBM MQ Explorer. For more information about the **strmqcfg** command, see *strmqcfg* in the IBM MQ online product documentation.

If you have multiple installations of IBM MQ, see [“Multiple installations of IBM MQ Explorer” on page 23.](#)

To trace IBM MQ Explorer, use the **runwithtrace.cmd**; see [“Troubleshooting” on page 262](#)

Multiple installations of IBM MQ Explorer

An installation of IBM MQ Explorer can be included with each installation of IBM MQ on your system. Each installation of IBM MQ Explorer is part of a separate installation of IBM MQ, and uses its own workspace.

If your system has multiple installations of IBM MQ Explorer, the only local queue managers shown on an installation of IBM MQ Explorer are those local queue managers associated with that installation.

Copying settings between installations of IBM MQ Explorer

IBM MQ Explorer stores settings in a workspace; this workspace, identified by the name of the installation of IBM MQ associated with it, is located in your home directory in *IBM/WebSphereMQ/workspace -*

Installation1. In this example, the name of the workspace associated with the workspace is Installation1.

At startup, IBM MQ Explorer detects whether any workspaces from previous installations of IBM MQ are present on your system. If a workspace from an installation earlier than IBM WebSphere MQ 7.1 is detected, without an associated product installation present, settings are automatically copied.

If there is at least one workspace available from earlier installations of IBM MQ and IBM MQ Explorer, you can choose whether to copy settings from one of these, or to start with a new workspace. At startup, a dialog presents you with a list of these available workspaces, each of which is identified by the name of the IBM MQ installation associated with it.

If you choose to copy settings from an earlier workspace, the settings copied are those used by IBM WebSphere MQ Explorer 7.1 only:

- IBM MQ Explorer Preferences
- IBM MQ Explorer Test Configurations
- Remote Queue Manager connections
- Queue Manager sets
- Schemes and Filters
- JMS Admin objects
- Service Definition repositories
- IBM Integration Bus data (if applicable)
- Managed File Transfer data (if applicable)
- SupportPac MSOP (if applicable)

These settings are not copied from the workspace:

- Any preferences unrelated to IBM MQ Explorer, IBM Integration Bus, or Managed File Transfer.
- Any projects unrelated to Service Definition repositories.
- IBM MQ Explorer Test Results.
- Any other Eclipse settings, including views and perspective changes.
- Any other SupportPacs or extensions to IBM MQ Explorer; these add-ons must be reinstalled if you want to use them with your new workspace.

Installing into Eclipse environments

Install IBM MQ Explorer into your own Eclipse environment or Eclipse-based product to help with developing Java applications or to develop your own extensions.

Before you begin

If you are using a version of Windows with User Account Control (UAC) such as Windows 7, UAC is enabled, and your Eclipse-based product is installed into a directory under UAC such as C:\Program Files, then you must run the Eclipse-based product with elevated authority (**Run as Administrator**) before you install the IBM MQ Explorer feature into it. When installed, the product can then be run without elevated authority.

Note: If you place the IBM MQ Explorer plug-ins in another Eclipse environment, to use the full set of CipherSuites, and to operate with certified FIPS 140-2 or Suite-B compliance, a suitable Java runtime environment (JRE) is required. IBM Java 7 Service Refresh 4, Fix Pack 2 or a higher level of the IBM JRE provides the appropriate support.

About this task

IBM MQ Explorer is built on Eclipse as a stand-alone application using the Eclipse Rich Client Platform. You can also install IBM MQ Explorer into your own Eclipse environment or an Eclipse-based product, to aid with developing Java applications with IBM MQ, or developing your own extensions to IBM MQ

Explorer. To do this, you need an environment that is based on Eclipse 4.4.2 that includes the Graphical Editing Framework (GEF) tools. For details of which Eclipse-based products are supported, see [System Requirements for IBM MQ](#).

Procedure

To install into a compatible Eclipse-based environment:

1. Click **Help** and then click **Install New Software** in the Eclipse environment.
2. Click **Add** and then click **Archive**, and then browse to the mqexplorer/eclipse directory inside the IBM MQ installation directory. Select the file MQExplorerSDK.zip.
3. Click **OK** after optionally typing a name for the local site.
4. A category of MQ Explorer is displayed. Expand this category and select **MQ Explorer** and optionally, the translations.
5. Click **Next** and follow the instructions. Then, click the button to restart Eclipse (or the Eclipse-based product).
If the installation fails due to a missing bundle, for example, org.eclipse.draw2d, you must install the Eclipse Graphical Editing Framework (GEF) tools.
6. IBM MQ Explorer is available as a separate perspective. To view, click **Open perspective**, and then click **Other**.

What to do next

For local queue managers, this procedure is sufficient if IBM MQ Explorer is being used to administer remote queue managers. If there are local queue managers to administer, you must run the Eclipse-based product with the required environment settings for your operating system. In addition, the Eclipse-based product must be a 64-bit application to match the 64-bit local queue managers.

Windows On Windows, set the PATH environment variable to include the bin64 and java/lib64 directories of your IBM MQ installation. You can use the **setmqenv** command to do this before you start the Eclipse-based product from the same command line. For example, if IBM MQ is installed in directory C:\Program Files\IBM\MQ, enter the following command:

```
C:\Program Files\IBM\MQ\bin\setmqenv -s
```

Linux On Linux x86_64 systems, set the LD_LIBRARY_PATH environment variable to include the java/lib64 and lib64 directories of your IBM MQ installation before you run the Eclipse-based product. For example, if IBM MQ is installed in /opt/mqm:

```
export LD_LIBRARY_PATH=/opt/mqm/java/lib64:/opt/mqm/lib64:$LD_LIBRARY_PATH
```

Related tasks

[“Using IBM MQ Explorer trace in other Eclipse environments” on page 263](#)

IBM MQ Explorer can be installed into your own Eclipse environment or Eclipse-based product to help develop Java applications, or to develop your own extensions.

Displaying context-sensitive help (pop-up window help)

At any time when you are working in IBM MQ Explorer, you can get context-sensitive help.

About this task

In wizards, the help is displayed in the banner of the wizard; in the rest of IBM MQ Explorer, the help is displayed in the pop-up window help. Pop-up window help can be displayed in two different ways, depending on where it is requested from, and what the default preferences have been set to. The two displayed forms are:



- Small, yellow boxes that contain a brief description, explanation, or instruction relevant to the part of the interface that currently has focus.
- A section that slides out of the side of the current dialog or perspective, that contains a brief description, explanation, or instruction relevant to the part of the interface that currently has focus.

The pop-up window help usually also contains a hyperlink that opens the help system to provide more detailed information.

The pop-up window help is available on a variety of objects in the IBM MQ Explorer interfaces; for example, folders, views, and properties dialogs.

To display the pop-up window help in IBM MQ Explorer:

Procedure

1. Bring focus to a part of the interface; for example, click a folder or hover over a properties dialog.
2. Display the pop-up window help:
 -  On Windows, press F1.
 -  On Linux, press Ctrl+F1.

Results

The pop-up window help displays.

What to do next

You can change the pop-up window help preferences by following this process: Click **Window > Preferences > Help**

The **Help Preferences** dialog opens.

Configuring IBM MQ using IBM MQ Explorer

In the Navigator view, you can use the Properties dialog to configure certain IBM MQ properties that apply to the whole installation. If necessary, you can also configure the properties of individual queue managers.

About this task

You can configure certain properties of IBM MQ that apply to the whole installation of IBM MQ on the computer. You can configure individual queue managers to override the IBM MQ properties if necessary.

To configure IBM MQ, complete the following steps.

Procedure

1. In the Navigator view, right-click IBM MQ , then click **Properties...** The Properties dialog opens.
2. In the Properties dialog, configure any of the following types of property as required:
 - General: Basic IBM MQ properties, such as the default location of queue managers on the computer.
 - Extended: More advanced IBM MQ properties, such as how EBCDIC newline characters are converted to ASCII.
 - Exits: Configure IBM MQ to use code modules (exits) that you have written yourself.
 - Default log settings: Change the location and type of IBM MQ logs.
 - ACPI: Specify how IBM MQ should respond when the computer tries to hibernate.
 - Alert monitor: Configure IBM MQ to alert you when there is a problem, such as a required queue that is missing.

Results

Any changes you make to the IBM MQ properties are made for all queue managers and objects on the computer, unless the individual queue managers are set up differently to override the IBM MQ settings.

Creating and configuring queue managers and objects



You can create, configure, and delete queue managers and objects in IBM MQ Explorer by using the Navigator view and Content view.

About this task

All of the queue managers and objects that you can administer in IBM MQ Explorer are contained in folders in the Navigator view. For example, the **Queue Managers** folder contains all of the queue managers that you can administer in IBM MQ Explorer; the **Channels** folder contains all of a queue manager's channels. For most folders, when you right-click the folder, a menu is displayed so that you can perform tasks such as creating a new object.

When you click a folder in the Navigator view, the contents of the folder are displayed in the Content view. For example, when you click the **Queues** folder, the queue manager's queues are displayed in the Content view. You can then right-click an object in the Content view to perform tasks such as opening the object's properties dialog, and deleting the object.

The following topics provide instructions about how to create, configure, and delete queue managers and objects in IBM MQ Explorer.

- [“Creating an IBM MQ object from a JMS object” on page 49](#)
- [“Configuring queue managers and objects” on page 51](#)
- [“Forcing changes to queue properties” on page 53](#)
- [“Strings in property dialogs” on page 655](#)
-  [“Configuring system parameters on z/OS queue managers” on page 57](#)
-  [“Configuring resource security on z/OS queue managers” on page 59](#)
- [“Deleting queue managers and objects” on page 55](#)

Objects in IBM MQ Explorer




In IBM MQ Explorer, all of the queue managers and their IBM MQ objects are organized in folders in the Navigator view.

You can perform the following tasks on many of the folders:

- To view the objects, click the relevant folder and the objects that are in the folder are listed in the Content view.
- To create new objects, right-click the folder.
- To configure or delete an object, right-click the object.

For more information about administering queue managers and their objects, see the following topics.

- [Queue managers](#)
- [Queues](#)
- [Topics](#)
- [Cluster topics](#)
- [Subscriptions](#)
- [Channels](#) (including client connections)
- [Listeners](#)
- [Process definitions](#)

- [Namelists](#)
- [Authentication information](#)
- [Trigger monitors](#)
- [Channel initiators](#)
- [Custom services](#)
-  [Storage classes](#)
-  [Queue sharing groups](#)
-  [Coupling facility structures](#)
- [Queue manager clusters](#)
- [JMS contexts](#)
- [JMS connection factories](#)
- [JMS destinations](#)

Related tasks

[“Configuring queue managers and objects” on page 51](#)

You can configure many of the attributes of queue managers and their objects from IBM MQ Explorer using the properties dialogs.

Queue managers

A queue manager is a program that provides messaging services to applications. Applications that use the Message Queue Interface (MQI) can put messages on queues and get messages from queues. The queue manager ensures that messages are sent to the correct queue or are routed to another queue manager.

The queue manager processes both the MQI calls that are issued to it, and the commands that are submitted to it (from whatever source). The queue manager generates the appropriate completion codes for each call or command.

The queue managers are the main components in an IBM MQ messaging network. The queue managers host the other objects in the network, such as the queues and the channels that connect the queue managers together. A queue manager must be running to perform the following tasks:

- Start channels
- Process MQI calls
- Create, delete, alter queues and channel definitions
- Run a command server to process MQSC commands

Related tasks

[“Creating and configuring queue managers and objects” on page 27](#)

You can create, configure, and delete queue managers and objects in IBM MQ Explorer by using the Navigator view and Content view.

[“Deleting queue managers and objects” on page 55](#)

When you delete a queue manager or an object in IBM MQ Explorer, the queue manager or object no longer exists on the system.

[“Showing or hiding a queue manager” on page 98](#)

By default, the Navigator view shows all of the queue managers on the computer on which IBM MQ Explorer is installed. However, if you have any queue managers that you are not currently administering, you can, if you wish, choose to hide them. You can also show and hide remote queue managers.

[“Removing a queue manager” on page 108](#)

You can remove a queue manager from IBM MQ Explorer if you no longer want to administer it in IBM MQ Explorer.

Related reference

[“Queue manager properties” on page 349](#)







You can set attributes for both local and remote queue managers.

IBM MQ queues

A queue is a container for messages. Business applications that are connected to the queue manager that hosts the queue can retrieve messages from the queue or can put messages on the queue.

A queue has a limited capacity in terms of both the maximum number of messages that it can hold and the maximum length of those messages.

Queue type	Description
Local queue	A local queue is a definition of both a queue and the set of messages that are associated with the queue. The queue manager that hosts the queue receives messages in its local queues.
Transmission queue	Transmission queues are a special type of local queue. When the queue manager sends a message to a queue on a remote queue manager, the transmission queue stores the message locally until the queue on the remote queue manager is available. To create a transmission queue, create a local queue and change its Usage attribute to Transmission .
Remote queue definition	Remote queue definitions are definitions on the local queue manager of queues that belong to another queue manager. To send a message to a queue on a remote queue manager, the sender queue manager must have a remote definition of the target queue.
Alias queue	Alias queues are not actually queues; they are additional definitions of existing queues. You create alias queue definitions that refer to actual local queues but you can name the alias queue definition differently from the local queue (the base queue). This means that you can change the queues that an application uses without needing to change the application; you just create an alias queue definition that points to the new local queue.
Model queue	A model queue is a template for queues that you want the queue manager to create dynamically as required. When an application tries to put a message on a model queue, the queue manager dynamically creates a local queue with the same name as the model queue. Queues that are created in this way can either be temporary or permanent.
Cluster queue	A cluster queue is a queue that has been shared in a cluster so that all of the queue managers in the cluster can put and get from the queue using cluster channels. For more information, see Queue manager clusters .

Queue type	Description
  Shared queue	z/OS only. A shared queue is a queue that has the queue sharing group disposition of Shared. All of the queue managers in the queue sharing group can put and get from the queue without needing active channels. Only local queues can have the disposition of Shared.  For more information, see Queue sharing groups .
  Group definition queue	z/OS only. A group queue is a queue that has the queue sharing group disposition of Group. Each of the queue managers in the queue sharing group has a copy of the queue (with the disposition Copy) stored on their own page set. Local, remote, alias, and model queues can have the disposition Group.  For more information, see Queue sharing groups .

For more information about queues, see [Queues](#) in the IBM MQ online product documentation.

Related concepts

[“JMS destinations \(queues and topics\)” on page 208](#)

A JMS destination is an object (a JMS queue or a JMS topic) that represents the target of messages that the client produces and the source of messages that the client consumes. In point-to-point messaging, destinations represent queues; in publish/subscribe messaging, destinations represent topics.

Related tasks

[“Creating and configuring queue managers and objects” on page 27](#)

You can create, configure, and delete queue managers and objects in IBM MQ Explorer by using the Navigator view and Content view.

[“Configuring queue managers and objects” on page 51](#)

You can configure many of the attributes of queue managers and their objects from IBM MQ Explorer using the properties dialogs.

[“Deleting queue managers and objects” on page 55](#)

When you delete a queue manager or an object in IBM MQ Explorer, the queue manager or object no longer exists on the system.

Related reference

[“IBM MQ queue properties” on page 412](#)

The attributes that you can set for a queue depend on the type of queue. Different types of IBM MQ queues have different properties. Some of the attributes do not apply to all types of queue, some attributes are specific to cluster queues, and some attributes are specific to z/OS queues.

Topics

A topic identifies what a publication is about. A topic is a character string that describes the subject of the information that is published in a Publish/Subscribe message. As a subscriber, you can specify a topic or range of topics by using wildcards to receive the information that you require.

A topic identifies what a publication is about and consists of a character string that can be up to 10,240 character long. Topics are key to the successful delivery of messages in a Publish/Subscribe system. Instead of including a specific destination address in each message, a publisher assigns a topic to each message. The queue manager matches the topic with a list of subscribers who subscribe to that topic, and delivers the message to each of those subscribers.

A publisher can control which subscribers receive a publication by choosing carefully the topic that is specified in the message.

The topic of a message does not have to be defined before a publisher can use it; a topic is created when it is specified in a publication or subscription for the first time.

For the latest about topic strings, wildcard characters, special characters, and topic trees, refer to the following information:

- A topic string can include any character from the Unicode character set, including the space character. However, there are characters that have special meanings: plus sign (+), number sign (#), asterisk (*), and question mark (?). For more information about these characters, see [Wildcard schemes](#) in the online product documentation.
- Topic strings are case-sensitive, and although a null character does not cause an error, do not use null characters in your topic strings. For the latest information about topic strings, see [Using topic strings](#) in the online product documentation.
- Each topic that you define is an element, or node, in the topic tree. For the latest information about topic trees, see [Topic trees](#) in the online product documentation.

Wildcards and special characters in topic strings

This topic details the wildcards that are used in the IBM WebSphere MQ 7.0 or later implementation of Publish/Subscribe messaging.

A topic can contain any character in the Unicode character set. However, the following three characters have a special meaning in IBM WebSphere MQ 7.0 or later Publish/Subscribe:

- The topic level separator "/".
- The multilevel wildcard "#".
- The single-level wildcard "+".

The topic level separator is used to introduce structure into the topic, and can therefore be specified within the topic for that purpose.

Wildcards are a powerful feature of the topic system in IBM MQ Publish/Subscribe. Wildcards allow subscribers to subscribe to more than one topic at a time. The multilevel wildcard and single level wildcard can be used for subscriptions, but they cannot be used within a topic by the publisher of a message.

However, if a publisher uses the characters "+" or "#" together with other characters in any topic level within a topic, these characters are not treated as wildcards, and they do not have any special meaning.

Related concepts

[“Publishers and subscribers” on page 117](#)

Publishers and subscribers are applications that send and receive messages (publications) using the publish/subscribe method of messaging. Publishers and subscribers are decoupled from one another so that publishers do not know the destination of the information that they send, and subscribers do not know the source of the information that they receive.

Related reference

[“Topic status attributes” on page 624](#)

The status attributes of topics.

Cluster topics

Topics can be clustered in a similar manner to cluster queues, although an individual topic object can be a member of only one cluster. A topic is made into a cluster topic by defining, on the topic object, the name of the cluster that is to host the topic, and the cluster routing mechanism to use for publications on this topic.

There are two options for routing publications across a publish/subscribe cluster: *direct routing* and *topic host routing*. To choose the message routing to use within the cluster, you set the **CLROUTE** property on the administered topic object to one of the following values:

- **DIRECT**
- **TOPICHOST**

By default, topic routing is **DIRECT**. This was the only option prior to IBM MQ 8.0. When you configure a direct routed clustered topic on a queue manager, all queue managers in the cluster become aware of

all other queue managers in the cluster. When performing publish and subscribe operations, each queue manager then connects directly to all the others.

From IBM MQ 8.0, you can instead configure topic routing as **TOPICHOST**. When you use topic host routing, all queue managers in the cluster become aware of the cluster queue managers that host the routed topic definitions. When performing publish and subscribe operations, queue managers in the cluster connect only to these topic host queue managers, and not directly to each other. The topic host queue managers are responsible for routing publications from queue managers on which publications are published to queue managers with matching subscriptions.

A topic host routed publish/subscribe cluster provides the following benefits:

- Improved scalability of larger clusters. Only the topic host queue managers need to be able to connect to all other queue managers in the cluster. Therefore, there are fewer channels between queue managers, and there is less inter-queue manager publish/subscribe administrative traffic than for direct routing. When subscriptions change on a queue manager, only the topic host queue managers need to be informed.
- More control over the physical configuration. With direct routing, all queue managers assume all roles, and therefore all need to be equally capable. With topic host routing, you explicitly choose the topic host queue managers. Therefore, you can ensure that those queue managers are running on adequate equipment, and you can use less powerful systems for the other queue managers.

The effect of defining a local topic as well as a cluster topic

You define a local topic object if you want publisher applications connected to a queue manager to publish only to locally connected subscribers. A local definition of a topic always overrides any clustered topic definitions on remote queue managers.

Note: You also need to specify a **Publication scope** of Queue manager on the local topic object. If **Publication scope** resolves to ALL, then remote subscribers are also sent publications published to the topic defined on this queue manager.

Multiple cluster topic definitions in a direct routed cluster

In a direct routed cluster, you do not usually define a cluster topic on more than one cluster queue manager. This is because direct routing makes the topic available at all queue managers in the cluster.

It is also not essential that the sole host queue manager is continually available, because the cluster topic definition is cached by the full repository queue managers and by all other queue managers in their partial cluster repositories. This caching allows for at least 60 days of availability while the host queue manager is unavailable.

If you need to alter a cluster topic definition, take care to modify it at the same queue manager it was defined on.

Multiple cluster topic definitions in a topic host routed cluster

In a topic host routed cluster, all publish/subscribe messaging is routed through the topic hosts. Therefore, to ensure scalability and availability, it is usual to define a cluster topic on more than one queue manager, and for the multiple cluster topic definitions to be identical.

Related concepts

[“Publishers and subscribers” on page 117](#)

Publishers and subscribers are applications that send and receive messages (publications) using the publish/subscribe method of messaging. Publishers and subscribers are decoupled from one another so that publishers do not know the destination of the information that they send, and subscribers do not know the source of the information that they receive.

Related reference

[“Topic properties” on page 458](#)

An IBM MQ topic is an IBM MQ object that identifies what a publication is about. You can set attributes for topics. Some topic attributes are specific to z/OS topics. Also, there are some attributes that you can alter only while you are creating a topic. You cannot modify these attributes after the IBM MQ topic has been created.

Subscriptions

A subscription is a record that contains the information about the topic or topics that the subscriber is interested in and wants to receive information about. Thus, the subscription information determines which publications get forwarded to the subscriber. Subscribers can receive information from many different publishers, and the information they receive can also be sent to other subscribers.

Published information is sent in an IBM MQ message, and the subject of the information is identified by a topic. The publisher specifies the topic when it publishes the information, and the subscriber specifies the topics on which it wants to receive publications. The subscriber is sent information about only those topics to which it subscribes.

IBM WebSphere MQ 7.0 or later queue managers use a Publish/Subscribe Engine to control the interactions between publishers and subscribers. The Publish/Subscribe Engine receives messages from publishers, and subscription requests from subscribers (to a range of topics). The Publish/Subscribe Engine's job is to route the published data to the target subscribers.

Subscribers can specify that they do not want to receive retained publications, and existing subscribers can ask for duplicate copies of retained publications to be sent to them. For more information on retained publications, see [“Publications” on page 33](#).

Related tasks

[“Configuring publish/subscribe for IBM WebSphere MQ 7.0 and later queue managers” on page 129](#)

In IBM MQ Explorer, you can configure IBM WebSphere MQ 7.0 and later queue managers as Publish/Subscribe Engines to route messages between publishing applications and subscribing applications. To test your configurations, you can register as a subscriber, and send and receive test publications if you are authorized to do so.

[“Viewing a list of subscribers” on page 138](#)

You can view a list of applications that are subscribed to topics on a Publish/Subscribe Engine, or a list of applications that are subscribed to a specific topic.

Related reference

[“IBM MQ subscription properties” on page 483](#)

You can set attributes for all types of subscriptions. Some of the attributes do not apply to all types of subscriptions, some attributes are specific to z/OS subscriptions.

[“Subscription status attributes” on page 626](#)

The status attributes of subscriptions.

Publications

Publications are messages that are sent by an application to the Publish/Subscribe Engine. The Publish/Subscribe Engine then sends the messages on to any applications that have subscribed to receive the messages.

The Publish/Subscribe Engine can handle publications that it receives in different ways, depending on the type of information contained in the publication.

State and event information

Publications can be categorized by the type of information that they contain:

State publications

State publications contain information about the current *state* of something, such as the price of stock or the current score at a soccer match. When something happens (for example, the stock price changes or the soccer score changes), the previous state information is no longer required because it is superseded by the new information.

A subscriber application wants to receive the current version of the state information on startup, and to be sent new information whenever the state changes.

Event publications

Event publications contain information about individual *events* that occur, such as a trade in some stock or the scoring of a particular goal. Each event is independent of other events.

A subscriber wants to receive information about events as they happen.

Retained publications

By default, when the Publish/Subscribe Engine has sent a publication to all interested subscribers, the Publish/Subscribe Engine deletes the publication. This type of processing is suitable for event information but is not always suitable for state information. A publisher can specify that the Publish/Subscribe Engine must keep a copy of a publication, which is then called a *retained publication*. The copy can be sent to subsequent subscribers who register an interest in the topic. This means that new subscribers do not have to wait for information to be published again before they receive it. For example, a subscriber who registers a subscription to a stock price would receive the current stock price straightaway, without waiting for the stock price to change (and, therefore, be re-published).

The Publish/Subscribe Engine retains only one publication for each topic, so the old publication is deleted when a new one arrives. So, ensure that only one publisher is sending retained publications on each topic.

Subscribers can specify that they do not want to receive retained publications, and existing subscribers can ask for duplicate copies of retained publications to be sent to them.

For more information about how to decide whether to use retained publications, see [Retained publications](#) in the IBM MQ online product documentation.

Related concepts

[“Publishers and subscribers” on page 117](#)

Publishers and subscribers are applications that send and receive messages (publications) using the publish/subscribe method of messaging. Publishers and subscribers are decoupled from one another so that publishers do not know the destination of the information that they send, and subscribers do not know the source of the information that they receive.

[“Topics” on page 30](#)

A topic identifies what a publication is about. A topic is a character string that describes the subject of the information that is published in a Publish/Subscribe message. As a subscriber, you can specify a topic or range of topics by using wildcards to receive the information that you require.

Channels

IBM MQ can use three different types of channels: a message channel, an MQI channel, and an AMQP channel.

Do not confuse these distinct types of channels:

Message channel

A message channel is a unidirectional communications link between two queue managers. IBM MQ uses message channels to transfer messages between the queue managers. To send messages in both directions, you must define a channel for each direction.

MQI channel

An MQI channel is bidirectional and connects an application (MQI client) to a queue manager on a server machine. IBM MQ uses MQI channels to transfer MQI calls and responses between MQI clients and queue managers.

Multi AMQP channel

An AMQP channel, which is bidirectional and connects an AMQP client to a queue manager on a server machine. IBM MQ uses AMQP channels to transfer AMQP calls and responses between AMQP applications and queue managers.

When referring to message channels, the word channel is often used as a synonym for a channel definition. It is usually clear from the context whether we are talking about a complete channel, which has two ends, or a channel definition, which has only one end.

Message channels

Message channel definitions can be one of the following types:

Message channel definition type	Description
Sender	A sender channel is a message channel that the queue manager uses to send messages to other queue managers. To send messages using a sender channel, you must also create, on the other queue manager, a receiver channel with the same name as the sender channel. You can also use sender channels with requester channels if you are implementing a "callback" mechanism.
Server	A server channel is a message channel that the queue manager uses to send messages to other queue managers. To send messages using a server channel, you must also create, on the other queue manager, a receiver channel with the same name as the server channel. You can also use server channels with requester channels. In that case, the requester channel definition at the other end of the channel requests the server channel definition to start. The server sends messages to the requester. The server can also initiate the communication as long as the server knows the connection name of the partner channel.
Receiver	A receiver channel is a message channel that the queue manager uses to receive messages from other queue managers. To receive messages using a receiver channel, you must also create, on the other queue manager, a sender or a server channel with the same name as this receiver channel.
Requester	A Requester channel is a message channel that the queue manager uses to receive messages from other queue managers. A Requester channel can request the partner channel defined at the remote end to start. If the partner channel is a Server channel, the Server channel accepts the start request and starts to send messages, from the transmission queue identified in the Server channel definition, to the Requester channel. If the partner channel is a Sender channel, the Sender channel accepts the start request but then closes the connection with the Requester. The Sender channel then starts, negotiates a session with the partner Requester channel and starts to send messages from the transmission queue identified in the Sender channel definition. This latter case essentially provides for a call back mechanism in that the Requester channel requests the Sender channel to call back.

Message channel definition type	Description
Cluster-sender	A cluster-sender (CLUSDR) channel definition defines the sending end of a channel on which a cluster queue manager can send cluster information to one of the full repositories. The cluster-sender channel is used to notify the repository of any changes to the queue manager's status, for example the addition or removal of a queue. It is also used to transmit messages. The full repository queue managers themselves have cluster-sender channels that point to each other. They use them to communicate cluster status changes to each other. It is of little importance which full repository a queue manager's CLUSDR channel definition points to. After the initial contact has been made, further cluster queue manager objects are defined automatically as required so that the queue manager can send cluster information to every full repository, and messages to every queue manager. For more information, see Queue manager clusters .
Cluster-receiver	A cluster-receiver (CLUSRCVR) channel definition defines the receiving end of a channel on which a cluster queue manager can receive messages from other queue managers in the cluster. A cluster-receiver channel can also carry information about the cluster-information destined for the repository. By defining the cluster-receiver channel, the queue manager indicates to the other cluster queue managers that it is available to receive messages. You need at least one cluster-receiver channel for each cluster queue manager. For more information, see Queue manager clusters .

For each channel you must define both ends so that you have a channel definition for each end of the channel. The two ends of the channel must be compatible types.

You can have the following combinations of channel definitions:

- Sender-Receiver
- Server-Receiver
- Requester-Server
- Requester-Sender (callback)
- Cluster-sender-Cluster-receiver

Message channel agents

Each channel definition that you create belongs to a particular queue manager. A queue manager can have several channels of the same or different types. At each end of the channel is a program, the message channel agent (MCA). At one end of the channel, the caller MCA takes messages from the transmission queue and sends them through the channel. At the other end of the channel, the responder MCA receives the messages and delivers them to the remote queue manager.

A caller MCA can be associated with a sender, server, or requester channel. A responder MCA can be associated with any type of message channel.

IBM MQ supports the following combinations of channel types at the two ends of a connection:

Caller		Direction of message flow	Responder	
Channel type	Listener required?		Listener required?	Channel type
Sender	No	Caller to Responder	Yes	Receiver
Server	No	Caller to Responder	Yes	Receiver
Server	No	Caller to Responder	Yes	Requester
Requester	No	Responder to Caller	Yes	Server
Requester	Yes	Responder to Caller	Yes	Sender

MQI channels

MQI channels can be one of the following types:

MQI channel type	Description
Server connection	A server connection channel is a bidirectional MQI channel that is used to connect an IBM MQ client to an IBM MQ server. The server connection channel is the server end of the channel.
Client connection	A client connection channel is a bidirectional MQI channel that is used to connect an IBM MQ client to an IBM MQ server. IBM MQ Explorer also uses client connections to connect to remote queue managers. The client connection channel is the client end of the channel. When you create a client-connection channel, a file is created on the computer that hosts the queue manager. You must then copy the client-connection file to the IBM MQ client computer.

AMQP channels



There is only one type of AMQP channel.

You use the channel to connect an AMQP messaging application with a queue manager, enabling the application to exchange messages with IBM MQ applications. An AMQP channel allows you to develop an application using MQ Light, and then deploy it as an enterprise application, taking advantage of the enterprise-level facilities provided by IBM MQ.

For more information on channels, see [Channels](#) in the IBM MQ online product documentation.

Related tasks

[“Creating and configuring queue managers and objects” on page 27](#)

You can create, configure, and delete queue managers and objects in IBM MQ Explorer by using the Navigator view and Content view.

[“Configuring queue managers and objects” on page 51](#)

You can configure many of the attributes of queue managers and their objects from IBM MQ Explorer using the properties dialogs.

[“Starting and stopping a channel” on page 93](#)

The way in which a channel is started depends on whether it is a caller channel or a responder channel. When you stop a channel, you can choose whether to stop the channel after the current batch of

messages has finished processing, or force the channel to shut down before the current message batch has finished processing.

[“Deleting queue managers and objects” on page 55](#)

When you delete a queue manager or an object in IBM MQ Explorer, the queue manager or object no longer exists on the system.

Related reference

[“Channel properties” on page 426](#)

You can set properties for all types of channels, including client-connection channels. Some properties are specific to certain types of channel.


Listeners

A listener is an IBM MQ process that listens for connections to the queue manager.

Each listener object in IBM MQ Explorer represents a listener process; however, if you start a listener process from the command line, the listener is not represented by a listener object in IBM MQ Explorer. Therefore, to administer the listener process from IBM MQ Explorer, create the listener object in IBM MQ Explorer. When you start the listener object in IBM MQ Explorer, the listener process starts.

There are different types of listener available in IBM MQ, depending on the transport protocol that the Message Channel Agent (MCA) uses to send and receive messages through the message channels:

- LU6.2
- TCP/IP
- NetBIOS
- SPX

 You can initiate new z/OS listeners in IBM MQ Explorer, which are displayed in the **Content** view, where they can be started and stopped. Only TCP/IP and LU6.2 are supported for z/OS listeners in IBM MQ Explorer.

For more information, see [Listeners](#) in the online IBM MQ product documentation.

Related tasks

[“Creating and configuring queue managers and objects” on page 27](#)

You can create, configure, and delete queue managers and objects in IBM MQ Explorer by using the Navigator view and Content view.

[“Configuring queue managers and objects” on page 51](#)

You can configure many of the attributes of queue managers and their objects from IBM MQ Explorer using the properties dialogs.

[“Deleting queue managers and objects” on page 55](#)

When you delete a queue manager or an object in IBM MQ Explorer, the queue manager or object no longer exists on the system.

Related reference

[“Listener properties” on page 455](#)

You can set attributes for all types of listeners. Some attributes are specific to certain types of listener.

Process definitions

A process definition contains information about the application that starts in response to a trigger event on a queue manager. When you enable triggering on a queue, you can create a process definition and associate it with the queue.

Each queue can specify a different process definition, or several queues can share the same process definition. If you create a process definition, the queue manager extracts the information from the process definition and places it in the trigger message for the trigger monitor to use.

If you want to trigger the start of a channel, instead of an application, you do not need to create a process definition because the transmission queue definition is used instead.

For more information, see [Process definitions](#) in the IBM MQ online product documentation.

Related concepts

[“Trigger monitors” on page 44](#)

A trigger monitor is an application that processes the trigger messages that are put on initiation queues when a trigger event occurs.

Related tasks

[“Creating and configuring queue managers and objects” on page 27](#)

You can create, configure, and delete queue managers and objects in IBM MQ Explorer by using the Navigator view and Content view.

[“Configuring queue managers and objects” on page 51](#)

You can configure many of the attributes of queue managers and their objects from IBM MQ Explorer using the properties dialogs.

[“Deleting queue managers and objects” on page 55](#)

When you delete a queue manager or an object in IBM MQ Explorer, the queue manager or object no longer exists on the system.

Related reference

[“Process definition properties” on page 490](#)

You can set attributes for process definitions. Some attributes do not apply to all types of process definitions. Some of the attributes are specific to z/OS process definitions.

Namelists

A namelist is an IBM MQ object that contains a list of names of other objects.

Typically, namelists are used by applications such as trigger monitors, where they are used to identify a group of queues, or with queue manager clusters to maintain a list of clusters referred to by more than one IBM MQ object. Namelists are also used to maintain lists of authentication information objects, which contain the authentication information about connections to LDAP servers.

For more information, see [Namelists](#) in the IBM MQ online product documentation.

Related concepts

[“Queue manager clusters” on page 48](#)

A cluster is a group of two or more queue managers that are logically associated and can share information with each other. Any queue manager can send a message to any other queue manager in the same cluster without you needing to set up a specific channel definition, remote queue definition, or transmission queue, because all of this information is held in the repository, to which all queue managers in the cluster have access.

[“Trigger monitors” on page 44](#)

A trigger monitor is an application that processes the trigger messages that are put on initiation queues when a trigger event occurs.

[“Authentication information” on page 40](#)

Authentication information objects contain connection details of servers that can be used to determine revocation status certificates.

Related tasks

[“Creating and configuring queue managers and objects” on page 27](#)

You can create, configure, and delete queue managers and objects in IBM MQ Explorer by using the Navigator view and Content view.

[“Configuring queue managers and objects” on page 51](#)

You can configure many of the attributes of queue managers and their objects from IBM MQ Explorer using the properties dialogs.

[“Deleting queue managers and objects” on page 55](#)

When you delete a queue manager or an object in IBM MQ Explorer, the queue manager or object no longer exists on the system.

Related reference









[“Namelist properties” on page 492](#)

You can set attributes for namelists. Some of the attributes are specific to z/OS namelists.

Authentication information

Authentication information objects contain connection details of servers that can be used to determine revocation status certificates.

An authentication information object contains authentication information that is used when checking whether a TLS certificate is revoked or not. The following table shows the IBM MQ TLS authentication information support for different platforms:

Table 1. Authentication information support for different platforms	
Platform	Support
 Windows  Windows IBM MQ on Windows systems	IBM MQ TLS supports checks for revoked certificates using OCSP, or using CRLs and ARLs on LDAP servers, with OCSP as the preferred method. IBM MQ classes for Java cannot use the OCSP information in a client channel definition table file. However, you can configure OCSP as described in Revoked certificates and OCSP in the product documentation.
 UNIX  UNIX IBM MQ on UNIX systems	IBM MQ TLS supports checks for revoked certificates using OCSP, or using CRLs and ARLs on LDAP servers, with OCSP as the preferred method. IBM MQ classes for Java cannot use the OCSP information in a client channel definition table file. However, you can configure OCSP as described in Revoked certificates and OCSP in the IBM MQ online product documentation.
 z/OS  z/OS IBM MQ on z/OS systems	IBM MQ TLS supports checks for revoked certificates using CRLs and ARLs on LDAP servers only. IBM MQ on z/OS systems cannot use OCSP.
 IBM i  IBM i IBM MQ on IBM i systems	IBM MQ TLS supports checks for revoked certificates using CRLs and ARLs on LDAP servers only. IBM MQ on IBM i systems cannot use OCSP.

For information about working with CRL & LDAP, see: [“Working with revoked certificates” on page 41.](#)

For information about working with OCSP, see: [“Working with Online Certificate Status Protocol \(OCSP\)” on page 42.](#)

For information about controlling access at a channel level, see [Channel authentication records](#) in the IBM MQ online product documentation.

Related concepts

[“Namelists” on page 39](#)

A namelist is an IBM MQ object that contains a list of names of other objects.

Related tasks

[“Creating and configuring queue managers and objects” on page 27](#)

You can create, configure, and delete queue managers and objects in IBM MQ Explorer by using the Navigator view and Content view.

[“Deleting queue managers and objects” on page 55](#)

When you delete a queue manager or an object in IBM MQ Explorer, the queue manager or object no longer exists on the system.

[“Configuring TLS on queue managers” on page 158](#)

After starting the IBM **strmqikm** (iKeyman) GUI, you can use it to manage TLS certificates.

Related reference









[“Authentication information properties” on page 494](#)

You can set attributes for all types of authentication information objects. Some of the attributes do not apply to all types of authentication information objects, and some attributes are specific to z/OS authentication information objects.

Working with revoked certificates

Authentication information objects contain connection details of responders or servers that can be used to determine the revocation status of certificates.

An authentication information object contains authentication information that is used when checking whether a TLS certificate is revoked or not. The following table shows the IBM MQ TLS authentication information support for different platforms:

Table 2. Authentication information support for different platforms	
Platform	Support
 Windows  Windows IBM MQ on Windows systems	IBM MQ TLS supports checks for revoked certificates using OCSP, or using CRLs and ARLs on LDAP servers, with OCSP as the preferred method. IBM MQ classes for Java cannot use the OCSP information in a client channel definition table file. However, you can configure OCSP as described in Revoked certificates and OCSP in the product documentation.
 UNIX  UNIX IBM MQ on UNIX systems	IBM MQ TLS supports checks for revoked certificates using OCSP, or using CRLs and ARLs on LDAP servers, with OCSP as the preferred method. IBM MQ classes for Java cannot use the OCSP information in a client channel definition table file. However, you can configure OCSP as described in Revoked certificates and OCSP in the IBM MQ online product documentation.
 z/OS  z/OS IBM MQ on z/OS systems	IBM MQ TLS supports checks for revoked certificates using CRLs and ARLs on LDAP servers only. IBM MQ on z/OS systems cannot use OCSP.
 IBM i  IBM i IBM MQ on IBM i systems	IBM MQ TLS supports checks for revoked certificates using CRLs and ARLs on LDAP servers only. IBM MQ on IBM i systems cannot use OCSP.

For more information, see [Securing IBM MQ](#) in the IBM MQ online product documentation.

Related concepts

[“Namelists” on page 39](#)

A namelist is an IBM MQ object that contains a list of names of other objects.

Related tasks

[“Creating and configuring queue managers and objects” on page 27](#)

You can create, configure, and delete queue managers and objects in IBM MQ Explorer by using the Navigator view and Content view.

[“Deleting queue managers and objects” on page 55](#)

When you delete a queue manager or an object in IBM MQ Explorer, the queue manager or object no longer exists on the system.

[“Configuring TLS on queue managers” on page 158](#)

After starting the IBM **strmqikm** (iKeyman) GUI, you can use it to manage TLS certificates.

Related reference

[“Authentication information properties” on page 494](#)









You can set attributes for all types of authentication information objects. Some of the attributes do not apply to all types of authentication information objects, and some attributes are specific to z/OS authentication information objects.

Working with Online Certificate Status Protocol (OCSP)

IBM MQ determines which Online Certificate Status Protocol (OCSP) responder to use, and handles the response received. You might have to take steps to make the OCSP responder accessible.

An authentication information object contains authentication information that is used when checking whether a TLS certificate is revoked or not.

Note: This information applies only to IBM MQ on UNIX and Windows systems. The following table shows the IBM MQ TLS authentication information support for different platforms:

<i>Table 3. Authentication information support for different platforms</i>	
Platform	Support
 Windows  Windows IBM MQ on Windows systems	IBM MQ TLS supports checks for revoked certificates using OCSP, or using CRLs and ARLs on LDAP servers, with OCSP as the preferred method. IBM MQ classes for Java cannot use the OCSP information in a client channel definition table file. However, you can configure OCSP as described in Revoked certificates and OCSP in the product documentation.
 UNIX  UNIX IBM MQ on UNIX systems	IBM MQ TLS supports checks for revoked certificates using OCSP, or using CRLs and ARLs on LDAP servers, with OCSP as the preferred method. IBM MQ classes for Java cannot use the OCSP information in a client channel definition table file. However, you can configure OCSP as described in Revoked certificates and OCSP in the IBM MQ online product documentation.
 z/OS  z/OS IBM MQ on z/OS systems	IBM MQ TLS supports checks for revoked certificates using CRLs and ARLs on LDAP servers only. IBM MQ on z/OS systems cannot use OCSP.
 IBM i  IBM i IBM MQ on IBM i systems	IBM MQ TLS supports checks for revoked certificates using CRLs and ARLs on LDAP servers only. IBM MQ on IBM i systems cannot use OCSP.

To check the revocation status of a digital certificate using OCSP, IBM MQ determines which OCSP responder to contact in one of two ways:

- Using the AuthorityInfoAccess (AIA) certificate extension in the certificate to be checked.
- Using a URL specified in an authentication information object or specified by a client application.

A URL specified in an authentication information object or by a client application takes priority over a URL in an AIA certificate extension.

The URL of the OCSP responder might lie behind a firewall; if so, reconfigure the firewall so the OCSP responder can be accessed or set up an OCSP proxy server. Specify the name of the proxy server by using the SSLHTTPProxyName variable in the SSL stanza. On client systems, you can also specify the name of the proxy server by using the environment variable MQSSLPROXY.

If you are not concerned whether TLS certificates are revoked, perhaps because you are running in a test environment, you can set OCSPCheckExtensions to NO in the SSL stanza. If you set this variable, any AIA certificate extension is ignored. This solution is unlikely to be acceptable in a production environment, where you probably do not want to allow access from users presenting revoked certificates.

The call to access the OCSP responder can return one of the following three outcomes:

Good

The certificate is valid.

Revoked

The certificate is revoked.

Unknown

This outcome can arise for one of three reasons:

- IBM MQ cannot access the OCSP responder.
- The OCSP responder has sent a response, but IBM MQ cannot verify the digital signature of the response.
- The OCSP responder has sent a response that indicates that it has no revocation data for the certificate.

By default, IBM MQ rejects a connection if it receives an OCSP response of Unknown, and issues an error message. You can change this behavior by setting the OCSPAuthentication attribute. This is held in the SSL stanza of the `qm.ini` file for UNIX systems, the WebSphere registry, or the SSL stanza of the client configuration file. It can be set using the IBM MQ Explorer on applicable platforms.

OCSP outcome Unknown

If IBM MQ receives an OCSP outcome of Unknown, its behavior depends on the setting of the OCSPAuthentication attribute. For queue managers, this attribute is held in the SSL stanza of the `qm.ini` file for UNIX systems, or the Windows registry, and it can be set using the IBM MQ Explorer. For clients, it is held in the SSL stanza of the client configuration file.

If an outcome of Unknown is received and OCSPAuthentication is set to REQUIRED (the default value), IBM MQ rejects the connection and issues an error message of type AMQ9716. If queue manager SSL event messages are enabled, an SSL event message of type MQRQ_CHANNEL_SSL_ERROR with ReasonQualifier set to MQRQ_SSL_HANDSHAKE_ERROR is generated.

If an outcome of Unknown is received and OCSPAuthentication is set to OPTIONAL, IBM MQ allows the SSL channel to start and no warnings or SSL event messages are generated.

If an outcome of Unknown is received and OCSPAuthentication is set to WARN, the SSL channel starts but IBM MQ issues a warning message of type AMQ9717 in the error log. If queue manager SSL event messages are enabled, an SSL event message of type MQRQ_CHANNEL_SSL_WARNING with ReasonQualifier set to MQRQ_SSL_UNKNOWN_REVOCATION is generated.

Digital signing of OCSP responses

An OCSP responder can sign its responses in any of three ways. Your responder will inform you which method is used.

- The OCSP response can be digitally signed using the same CA certificate that issued the certificate that you are checking. In this case, you do not need to set up any additional certificate; the steps you have already taken to establish SSL connectivity are sufficient to verify the OCSP response.
- The OCSP response can be digitally signed using another certificate signed by the same (CA) that issued the certificate you are checking. The signing certificate is flowed together with the OCSP response in this case. The certificate flowed from the OCSP responder must have an Extended Key Usage Extension set to id-kp-OCSPSigning so that it can be trusted for this purpose. Because the OCSP response is flowed with the certificate which signed it (and that certificate is signed by a CA which is already trusted for SSL connectivity) then no additional certificate setup is required.
- The OCSP response can be digitally signed using another certificate which is not directly related to the certificate you are checking. In this case, the OCSP Response is signed by a certificate issued by the OCSP responder itself. You must add a copy of the OCSP responder certificate to the key database of the client or queue manager which performs the OCSP checking. See [Adding a CA certificate \(or the CA part of a self-signed certificate\) into a key repository](#) in the IBM MQ online product documentation. When a CA certificate is added, by default it is added as a trusted root, which is the required setting in this context. If this certificate is not added, IBM MQ cannot verify the digital signature on the OCSP response and the OCSP check results in an Unknown outcome, which might cause IBM MQ to close the channel, depending on the value of OCSPAuthentication.

Related concepts

[“Namelists” on page 39](#)

A namelist is an IBM MQ object that contains a list of names of other objects.

Related tasks

[“Creating and configuring queue managers and objects” on page 27](#)

You can create, configure, and delete queue managers and objects in IBM MQ Explorer by using the Navigator view and Content view.

[“Deleting queue managers and objects” on page 55](#)

When you delete a queue manager or an object in IBM MQ Explorer, the queue manager or object no longer exists on the system.

[“Configuring TLS on queue managers” on page 158](#)

After starting the IBM **strmqikm** (iKeyman) GUI, you can use it to manage TLS certificates.

Related reference

[“Working with revoked certificates” on page 41](#)

Authentication information objects contain connection details of responders or servers that can be used to determine the revocation status of certificates.

[“Authentication information properties” on page 494](#)

You can set attributes for all types of authentication information objects. Some of the attributes do not apply to all types of authentication information objects, and some attributes are specific to z/OS authentication information objects.

Communication information object

IBM MQ Multicast offers low latency, high fanout, reliable multicast messaging. A communication information (COMMINFO) object is needed to use Multicast transmission.

Multicast is more efficient than traditional unicast publish/subscribe messaging as it can be scaled to a high number of subscribers without a detrimental effect in performance. IBM MQ enables reliable Multicast messaging by using acknowledgments, negative acknowledgments, and sequence numbers to achieve low latency messaging with high fanout.

IBM MQ Multicast fair delivery enables near simultaneous delivery, ensuring that no recipient gains an advantage. As IBM MQ Multicast uses the network to deliver messages, a publish/subscribe engine is not needed to fanout data. After a topic is mapped to a group address, there is no need for a queue manager, because publishers and subscribers can operate in a peer-to-peer mode. This process allows the load to be reduced on queue manager servers, and the queue manager server is no longer a potential point of failure.

The COMMINFO object contains the attributes associated with multicast transmission. To create COMMINFO object in the **Navigator** view:

1. In the **Navigator** view, expand the **Queue Managers** node.
2. Expand the node of the queue manager that you want to create a COMMINFO object on.
3. Right-click **Communication Information** to open the menu, then click **New > Multicast Communication Information** to open the COMMINFO wizard.
4. Work through the COMMINFO wizard to create COMMINFO object.

Trigger monitors

A trigger monitor is an application that processes the trigger messages that are put on initiation queues when a trigger event occurs.

If triggering is enabled for a queue and a trigger event occurs, the queue manager sends a trigger message to the initiation queue. The trigger monitor reads the trigger message and takes appropriate action, based on the data in the trigger message. Normally, this action would be to start some other application to process the queue that caused the trigger message to be generated. From the point of view of the queue manager, there is nothing special about a trigger monitor; it is just another application that reads messages from a queue (the initiation queue).

When you have started a trigger monitor, it just continues monitoring the specified initiation queue. You cannot stop a trigger monitor directly. When you stop the trigger monitor's queue manager, the trigger monitor stops too.

For more information, see [Trigger monitors](#) in the IBM MQ online product documentation.

Related concepts

[“Channel initiators” on page 45](#)

A channel initiator is an application that processes the trigger messages that are put on initiation queues when a trigger event occurs. A channel initiator is a special type of trigger monitor that starts channels rather than applications.

Related tasks

[“Starting a trigger monitor” on page 96](#)

To start a trigger monitor, you must first create a service that will start the trigger monitor.

Channel initiators

A channel initiator is an application that processes the trigger messages that are put on initiation queues when a trigger event occurs. A channel initiator is a special type of trigger monitor that starts channels rather than applications.

If triggering is enabled for a queue and a trigger event occurs, the queue manager sends a trigger message to the initiation queue. The channel initiator processes the trigger message and starts the channel. From the point of view of the queue manager, there is nothing special about a channel initiator; it is just another application that reads messages from a queue (the initiation queue).

Because a channel initiator is just a special type of trigger monitor, when you have started a channel initiator, it just continues monitoring the specified initiation queue. You cannot stop a channel initiator directly. When you stop the channel initiator's queue manager, the channel initiator stops too.

You also cannot create or delete a channel initiator. A channel initiator is created or deleted when its queue manager is created or deleted.

Related concepts

[“Trigger monitors” on page 44](#)

A trigger monitor is an application that processes the trigger messages that are put on initiation queues when a trigger event occurs.

Related tasks

[“Starting a channel initiator” on page 97](#)

To start a channel initiator, you must first create a service that will start the channel initiator.

Custom services

Custom services are services that you create to run commands automatically.

Custom services are stored in the **Services** folder on the queue manager to which the services belong. You can specify the command and other options that are run when the service starts and stops. You can automate a service to start and to run the command when the queue manager starts.

An example of when you might want to create a service is for starting a trigger monitor when the queue manager starts.

Related tasks

[“Creating and configuring queue managers and objects” on page 27](#)

You can create, configure, and delete queue managers and objects in IBM MQ Explorer by using the Navigator view and Content view.

[“Configuring queue managers and objects” on page 51](#)

You can configure many of the attributes of queue managers and their objects from IBM MQ Explorer using the properties dialogs.

[“Deleting queue managers and objects” on page 55](#)

When you delete a queue manager or an object in IBM MQ Explorer, the queue manager or object no longer exists on the system.

Related reference

[“Service properties” on page 468](#)

You can configure attributes for custom service objects in the Service properties dialog.

Storage classes

Storage classes can exist only on z/OS queue managers. A storage class maps one or more queues to a page set. This means that the messages on the queues are stored on the page set.

You can use storage classes to control where non-shared message data is stored for administrative, data set space and load management, or application isolation purposes. Only queues that are not shared store their messages on page sets. Therefore, shared queues do not use storage classes. The messages on shared queues are stored in coupling facility structures instead.

You can create storage classes in IBM MQ Explorer but the only storage class property that you can configure is the **Description** field.

Related tasks

[“Creating and configuring queue managers and objects” on page 27](#)

You can create, configure, and delete queue managers and objects in IBM MQ Explorer by using the Navigator view and Content view.

[“Deleting queue managers and objects” on page 55](#)

When you delete a queue manager or an object in IBM MQ Explorer, the queue manager or object no longer exists on the system.

[“Configuring queue managers and objects” on page 51](#)

You can configure many of the attributes of queue managers and their objects from IBM MQ Explorer using the properties dialogs.

Queue sharing groups

Queue sharing groups exist only on z/OS queue managers. A queue sharing group is a group of queue managers that can access the same shared queues. Each member of the queue sharing group has access to the same set of shared queues.

All the queue sharing groups to which the queue managers in IBM MQ Explorer belong are shown in the **Queue Sharing Group** folder. All the resources that members of a queue sharing group can access are shown in folders in the queue sharing group.

Queue sharing groups have a name of up to four characters. The name must be unique in your network, and must be different from any queue manager names.

You cannot create or configure queue sharing groups in IBM MQ Explorer.

For more information, see [Queue sharing groups and clusters](#) in the IBM MQ online product documentation.

Shared queues

A shared queue is a queue that has been defined on a queue manager in a queue sharing group and has the queue sharing group disposition of Shared. A shared queue's object definition is stored in the queue sharing group's shared repository on Db2®, and the messages on the shared queue are stored in a coupling facility structure on a physical coupling facility.

All of the queue managers in the queue sharing group can access the shared queue, which means that they can put and get messages on the shared queue without needing active channels. Because any queue manager can access the shared queue, an application is not dependent on the availability of any one queue manager.

All of the shared queues that belong to a queue manager are shown in the queue manager's folder. All of the shared queues in a queue sharing group are also shown in the queue sharing group's **Shared Queues** folder.

Group definitions

Group definitions is the collective term for IBM MQ objects that are defined on queue managers in a queue sharing group and have the queue sharing group disposition of Group. Any IBM MQ object that can be defined on a z/OS queue manager can have the queue sharing group disposition of Group. When you create a group definition object, the definition of the object is stored in the shared repository on Db2.

IBM MQ automatically creates a copy of the object (with queue sharing group disposition Copy) for each queue manager and stores it on the queue manager's page set zero with the queue manager's private objects, which have disposition Private. A page set is a data set that is specially formatted for use by IBM MQ. The messages on queues that have disposition Copy are also stored on page sets but they should not be stored on page set zero because if page set zero gets full, IBM MQ cannot function correctly. You can specify which page set the messages are stored on by creating one or more storage class objects which map the queues to page sets.

All of the group definitions that belong to a queue manager are shown in the queue manager's folder. All of the group definitions in a queue sharing group are also shown in the queue-sharing group's **Group Definitions** folder.

Related concepts

[“Coupling facility structures” on page 47](#)

The coupling facility objects in IBM MQ Explorer represent coupling facility structures on a physical coupling facility. Coupling facility structures store the messages that are on shared queues. Each coupling facility structure used by IBM MQ is dedicated to a specific queue sharing group, but a coupling facility can hold structures for more than one queue sharing group.

[“IBM MQ queues” on page 29](#)

A queue is a container for messages. Business applications that are connected to the queue manager that hosts the queue can retrieve messages from the queue or can put messages on the queue.

[“Storage classes” on page 46](#)

Storage classes can exist only on z/OS queue managers. A storage class maps one or more queues to a page set. This means that the messages on the queues are stored on the page set.

Coupling facility structures

The coupling facility objects in IBM MQ Explorer represent coupling facility structures on a physical coupling facility. Coupling facility structures store the messages that are on shared queues. Each coupling facility structure used by IBM MQ is dedicated to a specific queue sharing group, but a coupling facility can hold structures for more than one queue sharing group.

The physical coupling facility exists independently of the queue sharing group network, and typically runs on a different power supply. This means that the coupling facility is resilient to software failures and can be configured to be resilient to hardware failures and power-outages. The messages on the coupling facility structures are, therefore, highly available.

For more information, see [Managing the coupling facility](#) in the IBM MQ online product documentation.

Related concepts

[“Queue sharing groups” on page 46](#)

Queue sharing groups exist only on z/OS queue managers. A queue sharing group is a group of queue managers that can access the same shared queues. Each member of the queue sharing group has access to the same set of shared queues.

[“IBM MQ queues” on page 29](#)

A queue is a container for messages. Business applications that are connected to the queue manager that hosts the queue can retrieve messages from the queue or can put messages on the queue.

Related tasks

[“Creating and configuring queue managers and objects” on page 27](#)

You can create, configure, and delete queue managers and objects in IBM MQ Explorer by using the Navigator view and Content view.

[“Deleting queue managers and objects” on page 55](#)

When you delete a queue manager or an object in IBM MQ Explorer, the queue manager or object no longer exists on the system.

Related reference


[“Coupling facility structure properties” on page 513](#)

You can set attributes for coupling facility structures. Coupling facility structures are available only on z/OS.

Queue manager clusters

A cluster is a group of two or more queue managers that are logically associated and can share information with each other. Any queue manager can send a message to any other queue manager in the same cluster without you needing to set up a specific channel definition, remote queue definition, or transmission queue, because all of this information is held in the repository, to which all queue managers in the cluster have access.

For example, you can create a cluster, and then set a queue on one of its queue managers to be shared in the cluster. Now, on any other queue manager in the cluster, that queue is visible as if it were a local queue, and you can just open it and put a message directly on it.

 Note that sharing a queue in a cluster (a cluster queue) is different from sharing a queue in a queue sharing group (a shared queue) on z/OS queue managers. However, on z/OS, a cluster queue manager can also belong to a queue sharing group and can share its queue definitions with other queue managers in the queue sharing group.

Also, a queue manager on any platform can be a member of more than one cluster at the same time.

Cluster support also allows more than one queue manager to host an instance of the same queue (that is, a queue with the same name). This means that you can run more than one instance of an application, each receiving messages and running independently, thus spreading the workload between queue managers.

For more information, see [Distributed queuing and clusters](#) in the IBM MQ online product documentation.


Related concepts

[“Cluster repositories” on page 151](#)

A cluster repository contains information about the cluster; for example, information about the queue managers that are members of the cluster, and the cluster channels. Repositories are hosted by the queue managers in the cluster.

[“IBM MQ queues” on page 29](#)

A queue is a container for messages. Business applications that are connected to the queue manager that hosts the queue can retrieve messages from the queue or can put messages on the queue.

 [Queue sharing groups](#)

Queue sharing groups exist only on z/OS queue managers. A queue sharing group is a group of queue managers that can access the same shared queues. Each member of the queue sharing group has access to the same set of shared queues.

Creating an IBM MQ object from a JMS object

You can create new IBM MQ queues and topics based on your existing JMS queues and topics. The values of relevant properties of the JMS object are copied to the new IBM MQ object. In future, however, if you make a change to one of the objects, the changes are not reflected in the other object.

Before you begin

- [Add the initial context that contains the JMS queue or topic.](#)
- [Connect to the initial context.](#)

About this task

If the JMS object that you use to create an IBM MQ object specifies a queue manager name in its properties, you can only create the IBM MQ object on a queue manager with the same name. This means you might have to add a new queue manager with the name specified in the JMS object.

To create an IBM MQ object from an existing JMS object, complete the following steps.

Procedure

1. In the Navigator view, expand the initial context that contains the JMS object (either a JMS queue or a JMS topic), then click the **Destinations** folder to list the objects in the Content view.
2. In the Content view, right-click the object, then click **Create MQ Queue** or **Create MQ Topic** as appropriate.
The New Queue or New Topic wizard opens as appropriate.
3. In the wizard, click **Select**, then select the queue manager on which you want to create the new IBM MQ object.
The queue manager's name is displayed in the **Queue Manager** field of the wizard.
4. Work through the wizard to define the new IBM MQ object, then click **Finish**.

Results

The new IBM MQ object is created and displayed under the appropriate queue manager in IBM MQ Explorer.

What to do next

To view the new MQ object, in the Navigator view, expand the name of the queue manager on which you created the MQ object. You can now continue to configure the IBM MQ object as necessary.

To create an MQ object and a JMS object simultaneously, follow the instructions in: [“Creating an IBM MQ object and a JMS object simultaneously” on page 50](#) or [“Creating a JMS object and an IBM MQ object simultaneously” on page 216](#).

Related tasks

[“Creating and configuring queue managers and objects” on page 27](#)

You can create, configure, and delete queue managers and objects in IBM MQ Explorer by using the Navigator view and Content view.

[“Adding a queue manager from a JMS connection factory” on page 56](#)

You can add an existing queue manager to IBM MQ Explorer from a JMS connection factory that uses MQ MQI client transport (not bindings transport) and that specifies the host name and port that corresponds with the queue manager.

[“Creating a JMS object and an IBM MQ object simultaneously” on page 216](#)

When you create a new JMS object, you can optionally create a corresponding IBM MQ object of the same type.

[“Creating an IBM MQ object and a JMS object simultaneously” on page 50](#)

When you create a new IBM MQ object, you can optionally create a corresponding JMS object of the same type.

Creating an IBM MQ object and a JMS object simultaneously

When you create a new IBM MQ object, you can optionally create a corresponding JMS object of the same type.

Before you begin

- You must have an IBM MQ queue manager. If one does not exist, you can create one as described in: [“Creating and configuring queue managers and objects” on page 27](#)
- You must have a JMS initial context. If one does not exist, you can create one as described in: [Add the initial context that will contain the JMS topic](#)
- You must be connected to the JMS initial context as described in: [Connect to the initial context](#)

About this task

With the object creation wizards in IBM MQ Explorer, you can create an IBM MQ object and a JMS object simultaneously. You begin by launching your required object wizard, for example; an IBM MQ queue. Then you select the option to launch another wizard, for example; a JMS queue, after the object has been created. The second object wizard must be of the same object type, and the properties of one are mapped to the other.

Creating an IBM MQ queue and then creating a JMS queue simultaneously

About this task

When you create a new IBM MQ queue in IBM MQ Explorer, you can choose to launch the **New JMS Queue** wizard to create a JMS queue immediately after the IBM MQ **New Local Queue** wizard has finished. The **New JMS Queue** wizard now contains the details you entered when creating the IBM MQ queue.

To create a new IBM MQ queue and JMS queue simultaneously in IBM MQ Explorer, complete the following steps:

Procedure

1. Select the queue manager you want to add a new IBM MQ queue to in the Navigator view, and right-click on its **Queues** queue manager object folder.
2. Click **New > Local Queue** to open the **New Local Queue** wizard.
3. Type a name for your queue, then select **Start wizard to create a matching JMS Queue**. Continue through the wizard to create your queue.

Results

After you have completed the **New Local Queue** wizard, the **New Destination New JMS Queue** wizard opens, with many of the IBM MQ queue details mapped to the JMS queue.

Creating an IBM MQ topic and then creating a JMS topic simultaneously

About this task

When you create a new IBM MQ topic in IBM MQ Explorer, you can choose to launch the **New JMS Topic** wizard to create a JMS topic immediately after the IBM MQ **New Topic** wizard has finished. The **New JMS Topic** wizard now contains the details you entered when creating the IBM MQ topic.

To create a new IBM MQ topic and JMS topic simultaneously in IBM MQ Explorer:

Procedure

1. Select the queue manager you want to add a new IBM MQ topic to in the Navigator view, and right-click on its **Topics** queue manager object folder.
2. Click **New > Topic** to open the **New Topic** wizard.
3. Type a name for your topic, then select **Start wizard to create a matching JMS topic**. Continue through the wizard to create your topic.

Results

Once you have completed the **New Topic** wizard, the **New Destination New JMS Topic** wizard opens, with many of the IBM MQ topic details mapped to the JMS topic.

Related tasks

[“Creating a destination” on page 215](#)

A JMS client uses a destination object to specify the target of messages that the JMS client produces and the source of messages that the JMS client receives. Destination objects can represent queues (for point-to-point messaging) or topics (for publish/subscribe messaging).

[“Creating and configuring queue managers and objects” on page 27](#)

You can create, configure, and delete queue managers and objects in IBM MQ Explorer by using the Navigator view and Content view.

[“Creating an IBM MQ object from a JMS object” on page 49](#)

You can create new IBM MQ queues and topics based on your existing JMS queues and topics. The values of relevant properties of the JMS object are copied to the new IBM MQ object. In future, however, if you make a change to one of the objects, the changes are not reflected in the other object.

[“Creating a JMS object from an IBM MQ object” on page 217](#)

You can create new JMS administered objects based on your existing IBM MQ objects.

Related reference

[“Destination properties” on page 590](#)

You can view and set destination properties in the Destination properties dialog. The properties that are available in the dialog depend on the type of destination.

[“Connection factory properties” on page 561](#)

You can view and set connection factory properties in the Connection Factory properties dialog. The properties that are available in the dialog depend on which messaging provider the connection factory uses.

Configuring queue managers and objects

You can configure many of the attributes of queue managers and their objects from IBM MQ Explorer using the properties dialogs.

About this task



Attention: Security policies for AMS are not manageable by IBM MQ Explorer for IBM MQ for z/OS.

On the z/OS platform you must use CSQ0UTIL.

To configure a queue manager or object using the properties dialog, complete the following steps.

Procedure

1. In the Navigator view, click the relevant folder to list its contents in the Content view.

For example, if you want to configure a queue, click the **Queues** folder to list the queue manager's queues in the Content view.

2. In the Content view, right-click the queue manager or object, then click **Properties**.

The properties dialog for the queue manager or object opens.

3. Edit the properties as required.




4. To apply the changes without closing the dialog, click **Apply**, or to close the dialog and save your changes, click **OK**.

Results

You can see many of your changes immediately but some changes, for example, changing the default location of the queue manager's TLS key repository, do not take effect until you have stopped and restarted the queue manager.

Example

For more information about the properties of each type of object, see the following topics:

- [Queue manager properties](#)
- [Queue properties](#)
- [Channel properties](#)
- [Listener properties](#)
- [Queue manager manual set properties](#)
- [Queue manager automatic set properties](#)
- [Topic properties](#)
- [Service properties](#)
- [Subscription properties](#)
- [Process definition properties](#)
- [Namelist properties](#)
- [Authentication information properties](#)
-  [Storage class properties](#)
-  [Queue sharing group properties](#)
-  [Coupling facility structure properties](#)
- [Cluster queue manager properties](#)
- [Cluster queue properties](#)
- [Application Connection properties](#)
- [Message properties](#)
- [Connection factory properties](#)
- [Destination properties](#)

Related tasks

[“Creating and configuring queue managers and objects” on page 27](#)

You can create, configure, and delete queue managers and objects in IBM MQ Explorer by using the Navigator view and Content view.

[“Deleting queue managers and objects” on page 55](#)

When you delete a queue manager or an object in IBM MQ Explorer, the queue manager or object no longer exists on the system.

Related reference

[“Strings in property dialogs” on page 655](#)

To include certain characters in a string, you must punctuate the string in a particular way.

Forcing changes to queue properties

If the changes that you are making to the attributes of a queue affect the operation of a queue manager or another program, you might be asked to confirm whether you want to force the changes to the queue attributes.

About this task

In certain circumstances, when you change queue attributes and then click **OK**, a confirmation dialog is displayed that informs you that the changes you are making affect the operation of the queue manager or another program that is using the object. You are then asked if you want to force the change to the object's attributes. For the confirmation dialog to appear, you must have the object open in a separate application.

You must force changes to *local queues* in the following circumstances:

- The **Shareability** attribute on the **Extended** property page is specified as Not shareable.
- One or more applications have the queue open for input.
- Both of the following statements are true:
 - The **Usage** attribute is changed.
 - Either one or more messages are on the queue, or one or more applications have the queue open.

You must force changes to *alias queues* in the following circumstances:

- The **Base object** attribute is specified.
- An application has the queue open.

You must force changes to *remote queues* in the following circumstances:

- The **Transmission queue** attribute is changed.
- One or more applications have this queue open as a remote queue.
- Both of the following statements are true:
 - Any of **Remote queue**, **Remote queue manager**, or **Transmission queue** are changed.
 - One or more applications have the queue open which resolved through this definition as a queue manager alias.

Related tasks

[“Configuring queue managers and objects” on page 51](#)

You can configure many of the attributes of queue managers and their objects from IBM MQ Explorer using the properties dialogs.

Related reference

[“IBM MQ queue properties” on page 412](#)

The attributes that you can set for a queue depend on the type of queue. Different types of IBM MQ queues have different properties. Some of the attributes do not apply to all types of queue, some attributes are specific to cluster queues, and some attributes are specific to z/OS queues.

Comparing the properties of two objects

You can compare the properties of an object with another object of the same type; for example, compare a queue with another queue, a topic with another topic, or a channel with another channel.

About this task

You can compare two objects that are on the same queue manager, or on different queue managers.

To compare the properties of two objects:, complete the following steps.

Procedure

1. In the Content view, right-click the object that you want to compare, then click **Compare with...**
The Compare With dialog opens.
2. In the Compare With dialog, select the object to compare with:
 - To compare with an object on the same queue manager, select the name of the object that you want to compare with from the **With** container, and then browse for the queue manager or queue with which to compare it.
 - To compare with a queue on a different queue manager:
 - a. Select a queue manager from the **On Queue Manager** list.
 - b. Select the name of the object that you want to compare with from the **With** container.
 - If you are comparing queue managers instead of queues, the option to browse for a queue is not available.

Results

By default, the **show differences only** check box is selected so that only the properties that are different are listed. To show all of the properties of each queue, clear the **show differences only** check box.

Related reference

[“Properties” on page 339](#)

Use this information to find out about the properties that you can view and edit, including properties that apply to the whole IBM MQ installation and the properties of an individual IBM MQ object such as a queue, a queue manager, or a channel.

Pinging a channel to verify a connection

When you define a channel, you must define both ends of the channel correctly otherwise the channel does not work. You can test that you correctly defined a channel by sending data as a special message to the remote queue manager, and checking that the data is returned. The data is generated by the local queue manager.

About this task

You must ping from the sender or server end of the channel. You cannot ping a channel that is running; before you ping a channel, make sure that the channel is stopped or is not trying again.

To ping a channel, complete the following steps.

Procedure

In the Content view, right-click the sender or server channel definition, then click **Ping**.

Results

If the channel is correctly defined, a message is displayed saying: IBM MQ successfully sent data to the remote queue manager and received the data returned. (AMQ4006)

If the channel is not correctly defined, an error message is displayed describing why you could not ping the channel.

Related tasks

[“Configuring queue managers and objects” on page 51](#)

You can configure many of the attributes of queue managers and their objects from IBM MQ Explorer using the properties dialogs.

Related reference

[“Channel properties” on page 426](#)

You can set properties for all types of channels, including client-connection channels. Some properties are specific to certain types of channel.

Purging an AMQP channel

You can purge an AMQP channel to remove messages on queues on that channel.

About this task

To purge a channel, complete the following steps.

Procedure

1. In the Navigator view, click the **Channels** folder to display the channels in the Content view.
2. In the Content view, right-click the channel, then click **Purge**.

Starting a channel manually

About this task

You can start caller channels in IBM MQ Explorer. When you start responder channels in IBM MQ Explorer, you are actually changing the responder channel's state from Stopped to Inactive; the listener then changes the state from Inactive to Started. You must, therefore, start a listener on your computer if you are using responder channels.

To start a channel:

Procedure

1. In the Navigator view, click the **Channels** folder to display the channels in the Content view.
2. In the Content view, right-click the channel, then click **Start**.
3. In the Purge Channel window, optionally specify that channels associated with a particular client ID are purged.
4. Click **OK** to purge the channel.

Results

The channel is purged.

Deleting queue managers and objects

When you delete a queue manager or an object in IBM MQ Explorer, the queue manager or object no longer exists on the system.

About this task

Important: Before you delete a queue manager or an object, make sure that none of your applications need it. If you delete a queue manager, all the objects for the selected queue manager, such as queues or channels, are deleted as well.

If you want to keep the queue manager or object on the system but you do not want it to be displayed in IBM MQ Explorer, you can hide it. For more information, see [Hiding queue managers in IBM MQ Explorer](#), and [Filtering the objects displayed in IBM MQ Explorer](#).

To delete a queue manager or an object, complete the following steps.

Procedure

1. In the Navigator view, click the relevant folder to list its contents in the Content view. For example, if you want to delete a queue, click the **Queues** folder to list the queues for the selected queue manager in the Content view.
2. In the Content view, right-click the queue manager or object, then click **Delete**.
To delete multiple objects, hold down the Shift or Ctrl key, select the objects you want to delete, right-click the selected objects, then click **Delete**.
If you are deleting a queue and the queue contains messages, a dialog asks if you want to clear the messages first. You cannot delete a queue without clearing its messages first.
3. When you are prompted, click **Delete** to confirm that you want to delete the queue manager or object.

Results

The queue manager or object is deleted from the system and any applications that need the queue manager or object no longer work properly.

Related tasks

[“Creating and configuring queue managers and objects” on page 27](#)

You can create, configure, and delete queue managers and objects in IBM MQ Explorer by using the Navigator view and Content view.

[“Sending test messages” on page 89](#)

You can use a test message to check whether an application or a queue manager can put a message on a queue. You can also browse messages that are already on a queue or clear messages from a queue.

Adding a queue manager from a JMS connection factory

You can add an existing queue manager to IBM MQ Explorer from a JMS connection factory that uses MQ MQI client transport (not bindings transport) and that specifies the host name and port that corresponds with the queue manager.

Before you begin

- [Add the initial context that contains the JMS connection factory.](#)
- [Connect to the initial context.](#)

About this task

To add a queue manager from a JMS connection factory, complete the following steps.

Procedure

1. In the Navigator view, click the **Connection Factories** folder that contains the connection factory to display the connection factory in the Content view.
2. In the Content view, right-click the connection factory, then click **Add Queue Manager**.
IBM MQ Explorer tries to add the queue manager to the Queue Managers folder using the connection details in the connection factory.
3. When prompted, click **Yes**.

Results

The queue manager is added to the Queue Managers folder using the connection details that are specified in the connection factory. It is possible for the same queue manager to be shown more than once in the Queue Managers folder if each connection uses different connection details; for example, a local queue manager could be connected using 'localhost' as the host name, and it could also be connected using the IP address of the host as the host name.

What to do next

If you specify the queue manager's name with a * wildcard, you will be prompted that the determined queue manager could change each time the same connection factory is used.

If you specify the queue manager's name with a * wildcard and the connection fails, you will not be able to add the disconnected queue manager to the explorer, as the name will be undetermined.

It is not necessary for the JMS connection factory to specify the host name and port that corresponds with the queue manager, a client channel definition table (CCDT) can be used instead. For more information, see [Client channel definition table](#) in the IBM MQ online product documentation.

Related tasks

[“Creating and configuring queue managers and objects” on page 27](#)

You can create, configure, and delete queue managers and objects in IBM MQ Explorer by using the Navigator view and Content view.

[“Creating an IBM MQ object from a JMS object” on page 49](#)

You can create new IBM MQ queues and topics based on your existing JMS queues and topics. The values of relevant properties of the JMS object are copied to the new IBM MQ object. In future, however, if you make a change to one of the objects, the changes are not reflected in the other object.

Configuring system parameters on z/OS queue managers

If the default system parameter module that is supplied with IBM MQ does not contain the system parameters that you want, you can create your own system parameter module and customized macros.

About this task

A subset of a z/OS queue manager's attributes are loaded from the queue manager's system parameter module when the queue manager starts. The default system parameter module that is supplied with IBM MQ is called CSQZPARM. In the system parameter module are three macros:

- CSQ6SYSP, which controls the connection and tracing parameters
- CSQ6LOGP, which controls log initialization
- CSQ6ARVP, which controls archive initialization

If the default system parameter module does not contain the system parameters that you want, you can create your own system parameter module and customized macros using the supplied JCL sample. You cannot perform these customizations from IBM MQ Explorer because IBM MQ Explorer cannot connect to a queue manager that is not running. For more information, see [Customizing your queue managers](#) in the IBM MQ online product documentation.

You can, however, perform the following tasks in IBM MQ Explorer:

- [Viewing the system parameters](#)
- [Overriding system parameters while the queue manager is running](#)

Viewing the system parameters

When the z/OS queue manager starts, it loads its system parameter module which sets the queue manager's initial system parameter values. When the queue manager is running, you can monitor and administer it from IBM MQ Explorer and, therefore, view the queue manager's initial system parameter values.

Before you begin

Before you can perform this task, you must have already added the z/OS queue manager to IBM MQ Explorer and IBM MQ Explorer must be connected to the queue manager. For more information, see [Showing a remote queue manager](#) and [Connecting or disconnecting a queue manager](#).

About this task

To view a z/OS queue manager's system parameters:

Procedure

In the Navigator view, right-click the queue manager, then click the relevant menu item to view the initial parameter values that you are interested in:

- To view the log archive settings, click **Configuration > Archive**
- To view the log settings, click **Configuration > Log**
- To view the connection and tracing settings, click **Configuration > System**

Results

A dialog opens. In the dialog, the **Initial** table contains the values of the system parameters that were loaded from the system parameter module when the queue manager started.

Overriding system parameters while the queue manager is running

While the queue manager is running, you can change and temporarily override certain system parameter values. You can make these changes from IBM MQ Explorer.

Before you begin

Before you can perform this task, you must have already added the z/OS queue manager to IBM MQ Explorer and IBM MQ Explorer must be connected to the queue manager. For more information, see [Showing a remote queue manager](#) and [Connecting or disconnecting a queue manager](#).

About this task

To temporarily override system parameters:

Procedure

1. In the Navigator view, right-click the queue manager, then click the relevant menu item to view the initial parameter values that you are interested in:

- To view the log archive settings, click **Configuration > Archive**
- To view the log settings, click **Configuration > Log**
- To view the connection and tracing settings, click **Configuration > System**

A dialog opens. In the dialog, there is a table called **Set**. If you have already edited the system parameters since the queue manager started, the **Set** table contains the new parameter values.

2. In the dialog, click **Properties....**

A Properties dialog opens.

3. In the Properties dialog, edit the parameters that you want to override, then click **OK** to close the Properties dialog.

For more information about the individual parameters, see [Queue manager properties](#).

Results

The changes that you have made are shown in the **Set** table. The changes are temporary; the next time that the queue manager is restarted, the changes are lost because the values in the system parameter module are applied again.

Related tasks

[“Showing a remote queue manager” on page 99](#)

If you want to administer a remote queue manager, you must connect IBM MQ Explorer to the remote queue manager, so that the queue manager then appears in the Navigator view.

[“Configuring resource security on z/OS queue managers” on page 59](#)

For z/OS queue managers, you can activate or deactivate security for the whole queue manager (the subsystem). If security is active at the subsystem level, you can configure the security of the queue

manager's resources, and, if the queue manager belongs to a queue sharing group, you can configure security for the whole of the queue sharing group.

Related reference

[“Queue manager properties” on page 349](#)

You can set attributes for both local and remote queue managers.

Configuring resource security on z/OS queue managers

For z/OS queue managers, you can activate or deactivate security for the whole queue manager (the subsystem). If security is active at the subsystem level, you can configure the security of the queue manager's resources, and, if the queue manager belongs to a queue sharing group, you can configure security for the whole of the queue sharing group.

About this task

If subsystem security is active, when a user accesses an IBM MQ resource, the queue manager signs the user on to the queue manager. If the user does not access any IBM MQ resources on the queue manager for a predetermined period of time, the user's user ID is "timed out" and is signed out.

In IBM MQ Explorer, you can perform the following tasks:

- [“Viewing the queue manager's security settings” on page 59](#)
- [“Configuring the timeout period of user IDs” on page 59](#)

For more information, see [Securing IBM MQ](#) in the IBM MQ online product documentation.

Viewing the queue manager's security settings

There can be none, one, or more security switches present that determine the security of the queue manager. The switches can be set on or set off, and the setting of the switches is determined by the presence or absence of switch profiles. In IBM MQ Explorer, you can view but not configure the setting of the security switches.

Before you begin

Before you can perform this task, you must have already added the z/OS queue manager to IBM MQ Explorer and IBM MQ Explorer must be connected to the queue manager. For more information, see [Showing a remote queue manager](#) and [Connecting or disconnecting a queue manager](#).

About this task

To view the queue manager's current security settings:

Procedure

In the Navigator view, right-click the queue manager, then click **Configuration > Security**.

Results

The Security dialog opens. The **Security Switches** table displays all the security switches that are present, and are relevant to the queue manager. The table shows whether each security switch is set on or set off, and which profile determined this setting.

Configuring the timeout period of user IDs

If a user is authenticated to access a resource on the queue manager but then doesn't access any of the queue manager's resources for a predetermined length of time, the user's user ID is timed out. IBM MQ can make regular checks to determine whether a user ID has timed out. In IBM MQ Explorer, you can configure the length of the timeout period, and the frequency of checks to determine whether the timeout period has expired.

Before you begin

Before you can perform this task, you must have already added the z/OS queue manager to IBM MQ Explorer and IBM MQ Explorer must be connected to the queue manager. For more information, see [Showing a remote queue manager](#) and [Connecting or disconnecting a queue manager](#).

About this task

To configure the timeout period and frequency of checks:

Procedure

1. In the Navigator view, right-click the queue manager, then click **Configuration > Security**. The Security dialog opens.
2. In the Security dialog, click **Properties...**. The Properties dialog opens.
3. In the Properties dialog, edit the parameters that you want to change.

For example, if the Security timeout value is 30 and the Security interval value is 10, every 10 minutes IBM MQ checks user IDs and their associated resources to determine whether any have not been used for 30 minutes. If a timed-out user ID is found, that user ID is signed off within the queue manager. If any timed-out resource information associated with non-timed out user IDs is found, that resource information is discarded. If you do not want to time-out user IDs, set the Security interval value to zero. However, if the Interval value is zero, storage occupied by user IDs and their associated resources is not freed until you issue a REFRESH SECURITY or RVERIFY SECURITY command from the command line.

4. Click **OK** to close the Properties dialog.

Results

The changes are shown in the table in the Security dialog.

Related reference

[“Queue manager properties” on page 349](#)

You can set attributes for both local and remote queue managers.

Testing your object definitions for problems

You can use the IBM MQ Explorer tests to check your object definitions for errors and potential problems.

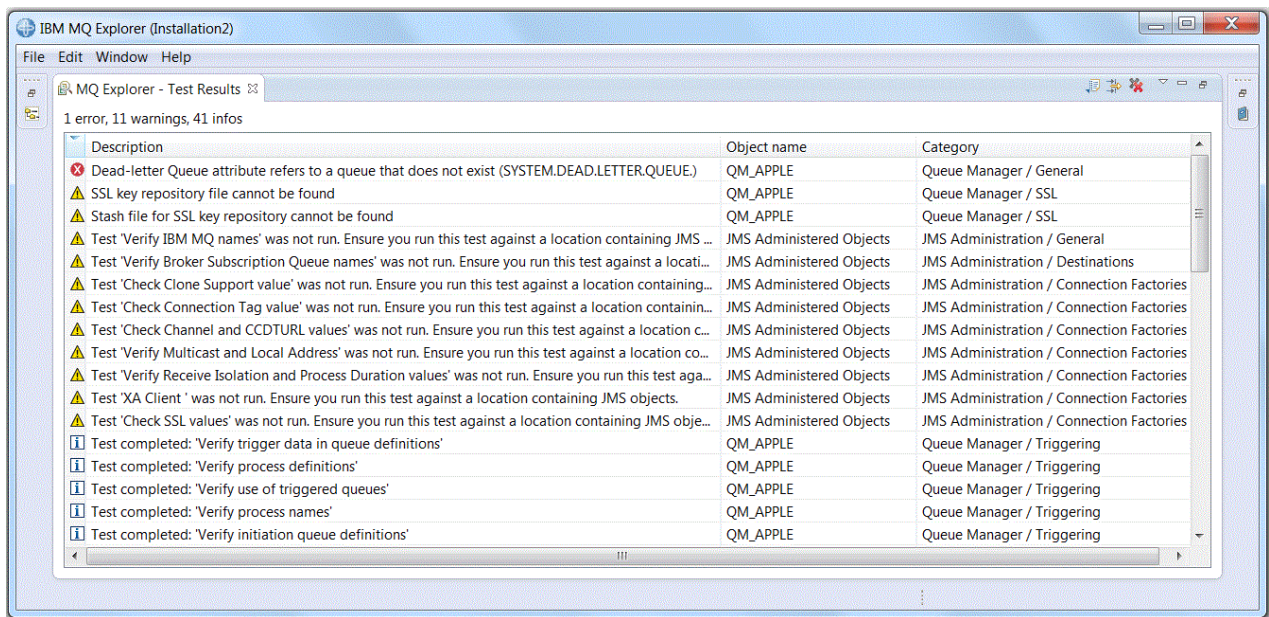
About this task

When you define objects in IBM MQ Explorer, certain properties of the objects are mandatory and you cannot create the objects without defining those properties. However, there are several properties that are not mandatory but which you should define so that your IBM MQ configuration works, so that the configuration is easier to maintain, or for audit purposes.

The IBM MQ Explorer tests check your object definitions for errors and potential problems. Each area of IBM MQ that can be checked is defined as a discrete test; for example, there is a test to check that you have defined matching channel pairs, a test to check that you do not have more than one TCP listener trying to listen on the same port, and a test to check that multiple queue managers on the same system are not using the same or similar names. The problems found by the core tests do not always indicate a serious error and sometimes just indicate where there is potential for confusion and mistakes when administering the objects; for example, two queue managers on the same system with similar names could cause problems.

A set of tests is supplied to check the main IBM MQ object definitions (for example, queues and channels); for the complete list of IBM MQ tests, see [IBM MQ tests](#). There are also other tests available that are supplied with other parts of IBM MQ Explorer, such as JMS administered objects.

Test results are displayed in the Test Results view, as shown in the following figure, which opens the first time that you run the tests.



You can extend the supplied set of tests to include your own custom tests so that IBM MQ Explorer can provide feedback that is directly relevant to how you use IBM MQ. For instructions and sample custom tests, see [Adding new tests](#).

Related tasks

[“Enabling installed plug-ins” on page 255](#)

If a new plug-in that you install in IBM MQ Explorer is not enabled by default, you can enable it by using the Preferences dialog.

[“Running tests” on page 61](#)

The tests in IBM MQ Explorer are run as test configurations. A test configuration contains a selection of tests and a list of objects (or types of object) against which the tests are run when you run the test configuration.

[“Adding new tests” on page 72](#)

You can extend the set of tests that is supplied with IBM MQ Explorer to include your own custom tests.

Running tests

The tests in IBM MQ Explorer are run as test configurations. A test configuration contains a selection of tests and a list of objects (or types of object) against which the tests are run when you run the test configuration.

About this task

There is a default test configuration for each object type which you can run directly from any of the objects or folders in the **Navigators** view. For more information, see [“Running the default tests” on page 62](#).

You can also create and edit your own test configurations to include new tests that you have written yourself or that you have obtained from a third party. For more information, see [“Creating and running your own test configuration” on page 62](#).

When you have run a test configuration, you can rerun an individual test without editing the test configuration. For more information, see [“Rerunning an individual test” on page 63](#).

Related tasks

[“Adding new tests” on page 72](#)

You can extend the set of tests that is supplied with IBM MQ Explorer to include your own custom tests.

[“Testing your object definitions for problems” on page 60](#)

You can use the IBM MQ Explorer tests to check your object definitions for errors and potential problems.

Running the default tests

The default test configuration contains the tests that are appropriate for the type of object against which you are running the test configuration.

About this task

You cannot change the selection of tests that are included in the default test configuration. If you edit the default test configuration, next time you run the default test configuration, the edited test configuration is not used; a new test configuration containing the default tests is created instead.

To run the default test configuration, complete the following steps.

Procedure

In the **Navigator** view, right-click the object or folder against which you want to run the tests, then click **Tests > Run Default Tests**.

While the tests are running, click **Run in Background** on the progress bar to run the tests in the background while you continue working. Alternatively, on the General page of the **Preferences** dialog, select the **Always run in background** check box. To view the progress of the tests while they run in the background, open the Progress view: click **Window > Show View > Other** then click **Basic > Progress**.

Results

When the test run has finished, a confirmation message is displayed. You can switch off this confirmation message in the **Preferences** dialog.

The first time that you run any tests, the **Test Results** view opens within the IBM MQ Explorer window. The test results are displayed in the **Test Results** view.

Related tasks

[“Creating and running your own test configuration” on page 62](#)

To have more control over the tests that are run or to include new tests that you have written, you can create and edit your own test configurations.

Creating and running your own test configuration

To have more control over the tests that are run or to include new tests that you have written, you can create and edit your own test configurations.

About this task

In a test configuration, you can select the tests that you want to run and also the objects or types of objects that you want to run the tests against. When you create a test configuration, the default set of tests is selected for the type of object from which you opened the dialog. However, you can change this selection and also add other types of object to the test configuration.

To create and run your own test configuration, complete the following steps.

Procedure

1. In the **Navigator** view, right-click an object or folder, then click **Tests > Run Custom Test Configuration**.

The **Run Tests Configurations** dialog opens.

2. In the **Run Tests Configurations** dialog, click **Tests** to select it.

The configuration icons become available.

3. In the **Run Tests Configurations** dialog, click **New** to create a test configuration. The default set of tests for the object or folder from which you opened the dialog is already selected in the new test configuration.
A new test configuration is added to the navigation tree. For example, if you opened the **Run Tests Configurations** dialog from Q1 queue on the QM1 queue manager, the Queues and Triggering categories of tests are already selected in the new test configuration; these tests are set to run only against queues on the QM1 queue manager.
4. In the **Name** field, type a meaningful name for the new configuration.
5. On the **Tests** page, select the tests, or categories of tests, to run when you run this test configuration.
6. If you want the test configuration to automatically update when you add new tests to IBM MQ Explorer, select the **Automatically include any new tests**.
7. On the **Objects** page, select the objects, or types of objects, to run the tests against when you run this test configuration.
8. If you want the test configuration to automatically update when you add new types of object definitions to IBM MQ Explorer, select the **Automatically include any new objects**.
9. Click **Apply** to save the new test configuration.
10. Click **Run** to run the new test configuration.

While the tests are running, click **Run in Background** on the progress bar to run the tests in the background while you continue working.

Results

When the test run has finished, a confirmation message is displayed. You can switch off this confirmation message in the **Preferences** dialog.

The first time that you run any tests, the **Test Results** view opens within the IBM MQ Explorer window. The test results are displayed in the **Test Results** view.

Related tasks

[“Adding new tests” on page 72](#)

You can extend the set of tests that is supplied with IBM MQ Explorer to include your own custom tests.

[“Running the default tests” on page 62](#)

The default test configuration contains the tests that are appropriate for the type of object against which you are running the test configuration.

Rerunning an individual test

If you have used the information in a test result to change objects in IBM MQ Explorer, you can rerun the test that produced that result without needing to run the whole test configuration again.

About this task

Rerunning an individual test enables you to quickly check whether the problem has been corrected.

Rerunning an individual test does not edit the test configuration and does not affect future test runs.

Procedure

To rerun an individual test: In the **Test Results** view, right-click the test result, then click **Run This Test Again**.

The test that generated the selected test result is run again and the test results generated by that test are updated in the **Test Results** view.

Related tasks

[“Running tests” on page 61](#)

The tests in IBM MQ Explorer are run as test configurations. A test configuration contains a selection of tests and a list of objects (or types of object) against which the tests are run when you run the test configuration.

Viewing test results

You can view test results in the **Test Results** view, which shows the results of the last test configuration run. You can filter or sort the test results that are displayed in the Test Results view.


About this task

The first time that you run tests against objects in IBM MQ Explorer, the **Test Results** view opens.

If you close the **Test Results** view, it reopens the next time you run a test. The view can be reopened manually at any time by clicking **Window > Show View > MQ Explorer - Test Results**.

Each row in the **Test Results** view represents a single test result. One test can generate one or more test results. To get more information about a test result, double-click the result. A new window opens to provide a brief explanation of why the test result was generated and whether you must take action.

The **Test Results** view always shows the test results of the last test configuration run. If you rerun an individual test, the original results of that test are replaced by the new results (or by nothing at all if the problems were resolved) but the rest of the original test results are retained.

Click Export Results  to save test results to a log file.

You can filter and sort the test results to make it easier to find the information that you need. For more information see [“Filtering test results in the Test Results view” on page 64](#) and [“Sorting test results in the Test Results view” on page 65](#).

Related tasks

[“Filtering test results in the Test Results view” on page 64](#)

You can filter the test results that are displayed in the **Test Results** view so that you can, for example, limit the number of results that are shown at one time, filter the results to show only the errors, or show only results that contain a specific string.

[“Sorting test results in the Test Results view” on page 65](#)

You can sort the test results in the **Test Results** view by specifying which column to sort by and whether to display the results in ascending or descending order.

[“Rerunning an individual test” on page 63](#)

If you have used the information in a test result to change objects in IBM MQ Explorer, you can rerun the test that produced that result without needing to run the whole test configuration again.


Filtering test results in the Test Results view

You can filter the test results that are displayed in the **Test Results** view so that you can, for example, limit the number of results that are shown at one time, filter the results to show only the errors, or show only results that contain a specific string.

About this task

To filter the test results displayed:

Procedure

1. In the **Test Results** view, click the filter icon  to open the **Filters** dialog.
The **Filters** dialog opens.
2. Edit the filters as required. For example, to show results that have names containing "IBM"; set **Object name** to contains, and type IBM in the field.
3. Click **OK** to apply your changes and close the dialog.

Results

The Test Results view is refreshed to show only the test results that match the filter criteria.

Any changes you make in this dialog are applied to all views that list problems.

Related tasks

[“Viewing test results” on page 64](#)

You can view test results in the **Test Results** view, which shows the results of the last test configuration run. You can filter or sort the test results that are displayed in the Test Results view.

[“Sorting test results in the Test Results view” on page 65](#)

You can sort the test results in the **Test Results** view by specifying which column to sort by and whether to display the results in ascending or descending order.

Sorting test results in the Test Results view

You can sort the test results in the **Test Results** view by specifying which column to sort by and whether to display the results in ascending or descending order.

About this task

Click the column name to sort the test results in descending order. Clicking the same column name again sorts the test results in ascending order. For example:

Procedure

1. In the **Test Results** view, click the column header called **Description** to sort the test results in descending order by description.
2. In the **Test Results** view, click the column header called **Description** again to sort the test results in ascending order by description.

Related tasks

[“Viewing test results” on page 64](#)

You can view test results in the **Test Results** view, which shows the results of the last test configuration run. You can filter or sort the test results that are displayed in the Test Results view.

[“Filtering test results in the Test Results view” on page 64](#)

You can filter the test results that are displayed in the **Test Results** view so that you can, for example, limit the number of results that are shown at one time, filter the results to show only the errors, or show only results that contain a specific string.

IBM MQ supplied tests

Tests that you can use to check your IBM MQ object definitions for problems are supplied with IBM MQ Explorer.

The following categories of tests are supplied with IBM MQ Explorer to check IBM MQ objects:

- [General tests](#)
- [Cluster tests](#)
- [Queue tests](#)
- [Channel tests](#)
- [Listener tests](#)
- [Triggering tests](#)
- [TLS tests](#)

The tests listed in the following tables are supplied with IBM MQ Explorer to check your IBM MQ object definitions for problems. There are other tests supplied with IBM MQ Explorer to check objects such as JMS administered objects for example; such tests are not included in the following table.

General

The following table lists the tests that check for general problems in your IBM MQ definitions.

Test	Action	Description
Check queue manager names	Verifies queue manager names for potential problems	This test checks queue manager names, looking for names that are similar enough to cause confusion; for example, the test checks for names that are identical apart from capitalization. The test also displays warnings for queue managers that are hosted on different machines but with identical names.
Dead-letter queue definitions	Checks queue managers for dead letter queues	This test displays a warning for any queue manager that does not have a dead-letter queue, and one or more errors for any queue manager that has Dead-Letter Queue attributes that are not valid; for example, the name of a queue that does not exist, or a queue that cannot be used as a dead-letter queue. The test displays a warning or error if any messages are found on the dead-letter queue because this can be a useful indication of a problem with the IBM MQ setup. The test also displays warnings if any channels have a maximum message length larger than the dead-letter queue size.
FFST error log	Checks whether error logs have been written to FFST directory on this machine	This test displays an error if any FFST logs have been written to on this machine.
Stopped queue managers	Checks to see if any queue managers are stopped	This test displays a warning for each queue manager that is stopped.
Verify default transmission queues	Verifies default transmission queues	This test displays errors for any invalid uses of the Default Transmission Queue attribute, including a missing queue or a queue with an value that is not valid in the Type attribute.

Clusters

The following table lists the tests that check for problems in your cluster definitions.

Test	Action	Description
Cluster fails to resolve queue manager name	Checks that clusters can successfully resolve all queue manager names	This test displays an error if any of the cluster membership entries have not been correctly resolved because the queue manager has not been successfully contacted.
Cluster-sender channels in Retrying state	Checks whether any of the manually defined cluster sender channels are still in Retrying state	This test displays an error if a cluster-sender channel is in the Retrying state.
Confirm cluster attributes are set	Checks that all cluster channels have a cluster value set	This test displays an error for any cluster-sender or cluster-receiver channels that do not have cluster (or cluster namelist) attributes set.
Duplicate cluster members	Checks whether any cluster memberships list the same queue manager more than once	This test displays a warning if any cluster membership list contains duplicate entries for a single queue manager.
Two full repositories	Checks that all clusters have at least two queue managers maintaining full repositories of the cluster	This test displays a warning if any cluster has only one full repository.
Verify cluster namelist definitions	Verifies the use of namelists in cluster definitions	This test verifies the cluster namelist attributes of queues, channels, and queue managers. The test displays errors if matching namelists cannot be found, or if a namelist is empty.
Verify cluster names	Checks cluster name attributes for potential problems	This test checks the cluster name attributes of queues, channels, and queue managers. The test checks for names which are similar enough to cause confusion; for example, names that are identical apart from capitalization.
Verify cluster queue instances	Verifies that all instances of a cluster queue have the same attributes	This test displays a warning if different instances of a cluster queue have different attributes.

Queues

The following table lists the tests that check for problems in your queue definitions.

Test	Action	Description
Identify full queues	Checks if any known queues are full	This test checks if the current depth of any known queue is equal to the value of the queue's Maximum Message Depth attribute.

Test	Action	Description
Verify alias queue definitions	Verifies alias queue definitions	This test checks the definitions of alias queues. The test checks the value of the Base Queue attribute of all alias queues found, and checks to see if the value is a valid target for the alias queue.
Verify queue names	Verifies names of MQ Queue objects	This test checks the names of queue definitions. The test checks for names that are similar enough to potentially cause confusion; for example names which are identical apart from capitalization.
Verify that queues are get-enabled	Verifies that all known queues are not get inhibited	This test verifies that all queues are get-enabled. Although it is not an error if a queue is not get-enabled, it might be useful to check for this when trying to identify the cause of unexpected behavior in your applications.
Verify that queues are put-enabled	Verifies that all known queues are not put inhibited	This test verifies that all queues are put-enabled. Although it is not an error if a queue is not put-enabled, it might be useful to check for this when trying to identify the cause of unexpected behavior in your applications.
Verify remote queue definitions	Verifies remote queue definitions	This test verifies the Remote Queue Manager and Remote Queue Name attributes of remote queue definitions.
Verify use of transmission queue in queues	Verifies the usage of transmission queues in remote queue definitions	This test checks the value of the Transmission Queue attribute in remote queue definitions. The test displays errors if the value is the name of a queue that does not exist or a queue of the wrong type.

Channels

The following table lists the tests that check for problems in your channel definitions.

Test	Action	Description
Identify in-doubt channels	Checks if any known channels are in an in-doubt status	This test displays a warning for any channel that is in doubt.

Test	Action	Description
Matching channel pairs	Verifies attributes on either end of a channel pair to look for potential problems	This test tries to find matching channel pairs. If the test finds the matching channel pairs, it checks that the two ends of the channel are of appropriate types and that the required attributes match at both ends of the pair. The test displays a warning message if no match or multiple matches are found for a channel; it displays an error if a channel pair has incompatible attributes.
Ping all non-running channels	Performs an MQ ping on all non-running sender, server and cluster-sender channels	This test pings all running sender, server, and cluster-sender channels that are not running and displays any unsuccessful responses in the Test Results view. Channels with Running status are not pinged because it is assumed that they have valid definitions.
Ping connection names	Verifies that all connection names referred to by channel definitions can be pinged	This test tries to ping the host names that are referred to in a channel definition's Connection name attributes. The test uses the ping utility that is provided by the operating system, if one is available, otherwise the test does nothing. The test displays a warning if a ping fails, and an error if the value of a required Connection name attribute is missing,
Resolve connection names	Verifies that all connection names referred to by channel definitions can be resolved	This test tries to resolve the host names that are referred to in channel definition's Connection name attributes, and displays a warning if the host name cannot be resolved to an IP address.
Verify channel interval values	Examines the ratios of the interval values on channel definitions	This test displays potential problems, such as channels with a heartbeat interval greater than the disconnect interval.
Verify channel names	Verifies names of MQ channel objects	This test checks the names of channel definitions. The test checks for names that are similar enough to cause confusion; for example, names that are identical apart from capitalization.

Test	Action	Description
Verify MCA User ID on server-connection channels	Verifies that all server-connection channels have a value entered for MCAUSER	This test displays warnings if any channels have missing MCA User ID attributes. Use it if you expect all server-connection channels to have MCA User ID set.
Verify use of transmission queues in channels	Verifies the usage of transmission queues in channel definitions	This test displays an error for any invalid use of the Transmission queue attribute in sender and server channel definitions, including missing queues, queues with invalid attributes, and transmission queues that are either not used by any channels or are used by multiple channels.

Listeners

The following table lists the tests that check for problems in your listener definitions.

Test	Action	Description
TCP listener port numbers	Checks the usage of TCP port numbers by channel listeners	This test validates the TCP port numbers used by channel listeners. The test displays warnings if invalid port numbers are used or if the same port is used by multiple queue managers.
Verify listener names	Verifies names of IBM MQ Listener objects	This test checks the names of listener object definitions. The test checks for names that are similar enough to potentially cause confusion; for example, names that are identical apart from capitalization.

Triggering

The following table lists the tests that check for problems in your triggering configuration.

Test	Action	Description
Verify initiation queue definitions	Verifies usage of initiation queue attribute of triggered queues	This test validates the Initiation Queue attribute of local and model queues. The test displays errors if the value specifies a local queue that cannot be found. The test also verifies that all the initiation queues have processes that have the queue open for input. If a queue does not have such a process, it indicates that there are no trigger monitors running against that queue.
Verify process names	Verifies names of IBM MQ process objects	This test checks the names of process definitions. The test checks for names that are similar enough to cause confusion; for example, names that are identical apart from capitalization.
Verify process definitions	Validates process object definitions	This test validates IBM MQ process definitions. The test checks that system processes that are specified in the object's Application ID attribute exist. Where the Application ID attribute does not give an absolute path, the test also displays a warning if multiple system processes with the given name can be found in the path environment.
Verify process definitions of queues	Verifies usage of process attribute of triggered queues	This test validates the Process Name attribute of local and model queues and displays errors for process names for which an IBM MQ process object definition cannot be found.
Verify trigger data queue definitions	Verifies usage of trigger data queue attribute of triggered queues	This test validates the Trigger Data attribute of local and model queues and displays errors for names for which a channel cannot be found.
Verify use of triggered queues	Verifies usage of trigger queues	If a queue meets its trigger conditions but the queue is not currently open for input, the test displays an error.

SSL/TLS

The following table lists the tests that check for problems in your SSL/TLS configuration.

Test	Action	Description
Verify that channels have been restarted	Verifies that SSL/TLS channels have been restarted since the last change to the SSL/TLS key repository	This test highlights any channels which have a last-started time earlier than the last modification time of the key repository, and which could therefore need refreshing.
Verify SSL channel authentication	Verifies that all channels require SSL/TLS authentication	This test highlights whether any channels have not got the CipherSpec attribute set. Use this test if you expect that all channels are using TLS.
Verify SSL client authentication	Verifies that all channels require SSL/TLS client authentication	This test highlights whether any channels have not got the Authentication of Parties Initiating Connections (SSLCAUTH) attribute set to Required. Use this test if you expect that all channels are using SSL/TLS and that all your clients will present a certificate for authentication.
Verify SSL key repository files	Verifies the presence of SSL/TLS key repositories	This test checks the SSL/TLS Key Repository attribute of queue managers and checks to see if a file can be found at that location. It also verifies that a password stash file can be found and is readable.
Verify SSL peer values	Verifies the SSL/TLS peer attributes used in channel definitions	This checks the Accept Only Certificates with Distinguished Names Matching These Values (SSLPEER) attribute of all known channels, reporting errors for invalid specifications, and warning when the value is used when the CipherSpec attribute is not.

Related tasks

[“Adding new tests” on page 72](#)

You can extend the set of tests that is supplied with IBM MQ Explorer to include your own custom tests.

Adding new tests

You can extend the set of tests that is supplied with IBM MQ Explorer to include your own custom tests.

About this task

The set of tests that is supplied with IBM MQ Explorer can be extended to include your own custom tests so that you can teach the IBM MQ Explorer to provide feedback that is directly relevant to your uses of IBM MQ.

The following topics provide more information about how to write your own tests:

- [Creating a new test](#). A step-by-step guide to preparing the Eclipse development environment for writing your own tests.
- [The 'WMQTest' interface](#). An explanation of the methods used in a basic test.
- [Design Considerations](#). Some pointers worth considering when writing your own tests.

Some sample source code is provided to assist with writing tests for IBM MQ Explorer:

- [Sample 1](#). A skeleton test that returns static data, as an example of the WMQTest interface.
- [Sample 2](#). A sample test that checks queue names against a defined naming convention, outputting errors if any queues are found that do not meet the standard.
- [Sample 3](#). A sample test that shows an asynchronous approach to requesting and processing data.
- [Sample 4](#). A diagnostic tool. Use this code in place of real test code to write to the console the objects that will be accessed by the real test code.

Creating a new test

You can create a new test to add to an existing category and set of tests.

About this task

These instructions describe how to create a new test in an existing category and set of tests (for example, the **Queues** test set in the **Queue manager tests** category) in IBM MQ Explorer. The instructions explain how to define the test in the Eclipse development environment. For information about writing the Java test source, see the [“WMQTest interface” on page 76](#).

If you want to create a new set or category of tests instead of using an existing set or category, or if you have created new objects to administer in IBM MQ Explorer and you are writing tests for the new objects, see [Creating new test categories, test sets, and object types](#).

- [Creating an Eclipse plugin project to contain the new test](#)
- [Defining a new test](#)
- [Writing a new test](#)
- [Deploying the new test](#)

Creating an Eclipse plug-in project to contain the new test

Before you begin

Ensure that you have installed the Eclipse Graphical Editing Framework (GEF) tools. For more information, see [“Installing into Eclipse environments” on page 24](#).

About this task

Create and configure a new plug-in project to contain your new test:

Procedure

1. Open the **Plug-in Development** perspective.
2. In the Package Explorer view, right-click, then click **New > Plug-in Project**. The New Plug-in Project wizard opens.
3. In the **Project name** field, type a name for the project that contains your new tests.
4. Click **Next**.
5. Edit the details in the **Version**, the **Name**, and the **Vendor** fields, and then click **Finish**.

Note the value in the **ID** field can be different from the value that you entered in the **Name** field on the previous page of the wizard. The project name is used only during development; the plugin ID is used by Eclipse to load and identify the plugin.

The new plug-in project is displayed in the Package Explorer view and the plug-in manifest file is automatically opened.

6. In the Plug-in Manifest editor, click the **Dependencies** tab. Two dependencies are already listed in the **Required Plug-ins** pane.

7. Add the following plug-ins to the **Required Plug-ins** pane:

- com.ibm.mq.explorer.tests
- com.ibm.mq.explorer.ui
- com.ibm.mq.pcf.event
- com.ibm.mq.runtime
- org.eclipse.core.resources

If the listed plug-ins are not available, install the Eclipse Graphical Editing Framework (GEF) tools. For more information, see [“Installing into Eclipse environments”](#) on page 24.

8. Save the MANIFEST.MF file.

Results

The plug-in project is ready to contain tests

Defining a new test

About this task

The following instructions describe how to define a new test an existing test set (for example, in the **Queues** test set), in the existing **Queue manager tests** category. For more information about creating new test sets in the Queue manager tests category, creating new categories, or defining new object types to be tested, see [Creating new categories, test sets, and object types](#).

Configure your plug-in to contain a new test:

Procedure

1. Ensure that the plugin.xml or MANIFEST.MF file is open in the Plug-in Manifest editor.

2. In the Plug-in Manifest editor, click the **Extensions** tab to display the **Extensions** page.

3. Click **Add...**

The New Extension wizard opens.

4. Highlight the **com.ibm.mq.explorer.tests.Tests** extension point, then click **Finish**.

The new tests extension is added to the **All Extensions** pane in the Plug-in Manifest editor.

5. Click the new test to highlight it, then enter the test's details as shown in the following table:

Attribute	Description	Example value
id	The unique identifier of the test.	com.ibm.mq.explorer.tests.samples.QueueNames
name	The meaningful name of the test.	My Queues Test
class	The Java class that contains the test. Do not enter this value yet; you can automatically enter this value when you create the class later.	com.ibm.mq.explorer.tests.samples.QueueNames

Attribute	Description	Example value
testset	The category to which the test belongs. The example value shown associates the test with the category Queue manager tests.	com.ibm.mq.explorer.tests.samples.wmq
testsubset	The subcategory to which the test belongs. The example value shown associates the test with the subcategory Queues.	queues
description	A description of what the test checks.	Checks queue names against simple naming conventions.
furtherinfo	The location of an HTML or XHTML document that contains more information about the test. This document is displayed in IBM MQ Explorer when you double-click the test in the Run Tests dialog or a test result in the Test Results view. For more information, see Test Documentation	doc/QueueNamesInfo.html (The location of the file relative to the plugin.xml file.)

6. Save the Plug-in Manifest editor file.

Results

The plug-in project is now configured to contain a new test; next you need to write the test itself.

Define a new test for each new test that you want to write.

Writing a new test

About this task

Create a new Java class that contains the test:

Procedure

1. In the **Extension Element Details** pane, click the label of the **class** field, which is underlined. The Java Attribute Editor wizard opens.
2. Ensure that only the **Inherited abstract methods** check box is selected, then click **Finish**. The Java class file opens in the Java editor.
3. Save the Plug-in Manifest editor file. Notice that the value in the class field is automatically inserted.
4. Edit the Java source.
5. Document the test in a valid XHTML or HTML file. Save the file with the name and location that is specified in furtherinfo attribute in the plugin.xml file. The location of the XHTML file might be local (stored in the same plug-in as the test; for example, in a doc subfolder) or remote (stored on a web server).

Results

You have completed writing the test and configuring the plugin that contains the test. Next, export the plug-in and deploy the plug-in to test it.

Write a new test for each test that you defined in the `plugin.xml` file.

Deploying the new test

About this task

Export the plug-in that contains your test (or set of tests) to the file system then restart IBM MQ Explorer so that the new plug-in is loaded and you can run the tests:

Procedure

1. In the Package Explorer view, right-click the plugin project, **com.ibm.mq.explorer.tests.samples**, then click **Export...**. The **Export...** dialog opens.
2. In the **Plug-in Development** perspective, click **Deployable plug-ins and fragments** to highlight it, then click **Next**.
3. In **Directory field**, enter the location of the IBM MQ Explorer Tests plugin. The location is `MQ_INSTALLATION_PATH\eclipse`, where `MQ_INSTALLATION_PATH` represents the high-level directory in which IBM MQ is installed.
4. Select your plug-in in **Available Plug-ins and Fragments**, and then click **Finish**.
5. Restart Eclipse, and switch to the IBM MQ Explorer perspective.

Results

You have deployed your new plugin. Now you can run your new tests.

WMQTest interface

Tests written for IBM MQ Explorer must belong to a Java class that extends the provided `WMQTest` class. This topic explains the interface and the operation of the provided methods.

- [Test attributes](#) - attributes for your test object
- [Creating the test](#) - the constructor for test objects
- [Test structure](#) - the beginning and end of the test
- [Running the test](#) - the main body for tests
- [User preferences](#) - accessing the preferences
- [Completing the test](#) - marking a test as complete
- [Creating a test result](#) - create test results
- [Dealing with canceling](#) - what happens if the user wants to cancel a test
- [Test documentation](#) - providing more information about the test

Test attributes

Define a test in the plug-in manifest file (`plugin.xml`) by using a collection of attributes. The attributes for a test are listed in the following table.

Attribute	Description
id	A string that provides a unique identifier for the test.
name	A meaningful name for the test.
class	The name of the Java class that contains the test source code.

Attribute	Description
testset	A string that defines the group in which to display the test; for example, wmq, which displays the test in the Queue manager tests category.
testsubset	A string that defines the sub-group in which to display the test; for example, queues, which displays the test in the Queues category.
description	A short description that describes what the test does.
furtherinfo	The location of an HTML or XHTML document that contains more information about the test. This document is displayed in IBM MQ Explorer when you double-click the test in the Run Tests dialog or a test result in the Test Results view.

You specify the values of these attributes in the plugin.xml file to define the test. These attributes can also be accessed programmatically using the WMQTest methods listed in the following table.

Method	Description
getTestID()	Returns the test ID.
getTestName()	Returns the name of the test.
getDescription()	Returns the description of the test.
getTestSet()	Returns a handle for the test set object that was created to be a parent for the test.
getFurtherInfoPath()	Returns the location of the XHTML or HTML document that contains more information about the test.

Creating the test

The IBM MQ Explorer Tests engine instantiates the test object using the provided constructor `WMQTest()`. There is no need to subclass this constructor.

Test structure

The `WMQTest` method `runTest` defines the body of the test, and is called to start a test running.

The end of the `runTest` method does not imply the end of the test; you must explicitly specify the end of the test using the `testComplete` method. You can implement tests so that they get the object data asynchronously.

The `runTest` method submits a request to get data about objects and the test runs from the listener method that receives the reply. This enables the test to wait for data without you needing to implement thread waiting; this is demonstrated in [Sample 3](#).

If a manual wait (sleep) is needed as a part of a test, you can use the object monitor for the test object to use the Java `wait` and `notify` methods. The threading of the test engine is implemented without using the object monitors of individual test objects.

Running the test

The IBM MQ Explorer Tests engine calls `runTest(WMQTestEngine, IProgressMonitor, contextObjects, treeNode)` to start the test running. The main body of your test must be here.

WMQTestEngine

The **WMQTestEngine** parameter provides a handle to the test engine that is running the test.

This is provided to allow tests to return results while a test is in progress using the test engine's `returnResult(WMQTestResult[], WMQTest)` method.

The first parameter of this method (`WMQTestResult[]`) contains the results to be returned, and the second parameter (`WMQTest`) must be 'this', so that the test engine knows where the results have come from. Using the **WMQTestEngine** parameter to return interim results is optional - alternatively, test results can be returned on test completion (see [Completing the test](#)).

IProgressMonitor

The **IProgressMonitor** parameter provides a handle to the GUI feedback monitor being used for the current test run. This allows your test to provide both textual feedback on the task and subtasks currently running, and a progress bar for current completion.

The handle to the Progress Monitor is cached by the default implementation of `runTest`, so if this has been used, a handle to the Progress Monitor can also be accessed using the `WMQTest` method `getGUIMonitor()`.

The Progress Monitor is a core Eclipse resource. See the [Eclipse API documentation](#) on the Web for further advice on using it.

contextObjects

The **contextObjects** parameter provides an `MQExtObject` array. The parameter provides the context of the test to be run so that the relevant check boxes are pre-selected when the user opens the Run Tests dialog.

treeNode

The **treeNode** parameter records which folder or object in the Navigator view was clicked to run the default tests or to open the Run Tests dialog.

User preferences

Tests must conform to the user preferences provided using the Eclipse Preferences dialog. Use the following methods to access the preferences:

- `PreferenceStoreManager.getIncludeHiddenQmgrsPreference()` which returns `true` if you include queue managers that have been hidden in IBM MQ Explorer in the test, or `false` if they must be excluded.
- `PreferenceStoreManager.getIncludeSysObjsPreference()` which returns `true` if system objects (objects which have names beginning with `SYSTEM.`) must be included in the test, or `false` if they must be excluded.

Completing the test

Complete a test by calling `testComplete(WMQTestResult[])`, passing it an array of test result objects. See [“Creating a test result”](#) on page 79 for guidance on test result objects.

You can return results at completion using this method in addition to, or as an alternative to, returning test results during a test run (as explained in [Running the test](#)). However, any results that are returned twice are displayed twice.

Even if your test uses the `WMQTestEngine` method `returnResult` to return all of its results, it must still call `testComplete` on completion. This is necessary to complete the test processing. You can provide an empty array of `WMQTestResult` objects in the `testComplete` method if there are no new results to be returned.

For more information, see [“Test structure” on page 77](#).

Creating a test result

Test results are implemented as `WMQTestResult` objects. Create results using:

WMQTestResult(int severity, String description, String qmgrname, String objectType)

where:

- `severity` is an integer identifying the severity of the problem. Use one of the following severity levels: `IMarker.SEVERITY_ERROR`, `IMarker.SEVERITY_WARNING` or `IMarker.SEVERITY_INFO`
- `description` is the string explaining the problem found by the test, to be displayed in the Problems View.
- `qmgrname` is the name of the queue manager where the problem was found.
- `objectType` is string giving the class of object where the problem can be found, for example, "Queues" or "Channels".

For more information about what to do with the test result object when it has been created, see [“Completing the test” on page 78](#).

Dealing with canceling

You can cancel the test run while it is running. Use the method `isCancelled()` to check if a test must stop.

A good test must regularly check whether it has been canceled to avoid delaying a user unnecessarily.

If you try to cancel a test but the test fails to respond for an extended period of time, the test engine forces the test to stop by ending the thread that is running the test. Do not rely upon this method, it is preferable that a test responds in time allowing the test to clean up any resources it has used, and to return any test results that have been generated so far.

Test documentation

You can provide additional documentation to explain the results that they return, and provide guidance on what must be done to resolve the problem.

Provide documentation in HTML, with the location identified in the `plugin.xml` file for the plug-in providing the test. For details about defining tests in XML, see [“Creating a new test” on page 73](#).

The location of the documentation HTML file can be:

- **internal** - Stored in the plug-in project providing the test itself. The location must be defined in the XML relative to the `plugin.xml` file itself. For example, `doc/TestDoc.html`
- **external** - Stored on a web-server, allowing maintenance of the documentation separately from the test itself. The location must be defined as a complete URL, beginning with 'http://'.

Creating new test categories, test sets, and object types

You can create new test categories, new test sets in an existing category, and new subsets in an existing test set. You can also define new object types for which you want to create tests.

About this task

All of the tests that are supplied with IBM MQ Explorer are grouped in the **Queue manager tests** category. In the **Queue manager tests** category, each test is associated with a specific test set; for example, **Queues** or **Channels**. The test sets are used to make the default selections in the **Run Tests** dialog, which is based on the type of folder or object in the **Navigator** view from which you opened the **Run Tests** dialog. The test sets are also used to specify which tests are run in the default set of tests.

You can see these categories and test sets if you open the **Run Tests** dialog (right-click a folder in the Navigator view, then click **Tests > Run custom test configuration**) and look at one of the test configurations on the **Tests** page of the dialog.

You can create new categories (like the **Queue manager tests** category). You can also create new test sets (like the **Queues** test set) in a category, and even new subsets in an existing test set.

If you create new object types and folders to display in the **Navigator** view of IBM MQ Explorer and you want to create tests that verify definitions of the new object types, you can define the new object types so that they are displayed as options on the **Objects** page of the **Run Tests** dialog.

For instructions on creating new tests in an existing test set in the Queue manager tests category, see [Creating a new test](#). The following instructions describe how to create new categories and test sets, and define new object types:

- [Creating a new test set in an existing category](#) (`com.ibm.mq.explorer.tests.Testset`)
- [Creating a new category and test set](#) (`com.ibm.mq.explorer.tests.TestCategories`)
- [Defining a new object type to be tested](#) (`com.ibm.mq.explorer.tests.ContextGroup`)

Do the following tasks in the **Plug-in Development** perspective.

Creating a new test set in an existing category (`com.ibm.mq.explorer.tests.Testset`)

About this task

To create a new test set in an existing category (a category that you did not create; for example the **Queue manager tests** category):

Procedure

1. On the **Extensions** page of the `plugin.xml` file, add the **`com.ibm.mq.explorer.tests.Testset`** extension to the **All Extensions** pane.
2. Configure the new test set according to the details in the following table:

Attribute	Description	Example value
categoryId	The unique identifier of the category in which you are creating the new test set.	<code>com.ibm.mq.explorer.tests.coretests.wmq</code>
id	The unique identifier of the category that you are creating.	<code>com.ibm.mq.explorer.tests.samples.NewCategory</code>
name	A meaningful name for the category.	My New Category
description	A brief description of the category.	This is my first new category.
icon	An optional icon that can be used to represent the category.	<code>icons/newcat.gif</code> (The location of the icon file relative to the <code>plugin.xml</code> file.)
furtherinfo	The location of an HTML or XHTML document that contains more information about the test. This document is displayed in IBM MQ Explorer when you double-click the test in the Run Tests dialog or a test result in the Test Results view.	<code>doc/MyObject.html</code> (The location of the HTML or XHTML file relative to the <code>plugin.xml</code> file.)

3. Save the `plugin.xml` file.

Results

You have created a new test set in an existing category.

Creating a new category and test set (com.ibm.mq.explorer.tests.TestCategorys)

About this task

If you are creating a new category, you can create test sets in that category by using a single extension; that is, you do not need to use the separate com.ibm.mq.explorer.tests.Testset extension as well.

To create a new category:

Procedure

1. On the **Extensions** page of the plugin.xml file, add the **com.ibm.mq.explorer.tests.TestCategorys** extension to the **All Extensions** pane.
2. Configure the new category according to the details in the following table:

Attribute	Description	Example value
id	The unique identifier of the category that you are creating.	com.ibm.mq.explorer.tests.samples.NewCategory
name	A meaningful name for the category.	My New Category
description	A brief description of the category.	This is my first new category.
icon	An optional icon that can be used to represent the category.	icons/newcat.gif (The location of the icon file relative to the plugin.xml file.)
furtherinfo	The location of an HTML or XHTML document that contains more information about the test. This document is displayed in IBM MQ Explorer when you double-click the test in the Run Tests dialog or a test result in the Test Results view.	doc/MyObject.html (The location of the HTML or XHTML file relative to the plugin.xml file.)

3. Save the plugin.xml file.

Results

You have created a new category.

What to do next

To create a new test set in this category:

1. Right-click the category, then click **New > testset** to add a new test set to the **All Extensions** pane.
2. Configure the new test set according to the details in the table in *Creating a new test set in an existing category*. Notice that you do not set a **categoryID** attribute because you are creating the test set in the category that you just created.
3. Save the plugin.xml file.

You have created a new test set in the new category.

Defining a new object type to be tested (*com.ibm.mq.explorer.tests.ContextGroup*)

About this task

If you have created new types of objects to be displayed in the Navigator view of IBM MQ Explorer and you want to create tests to check definitions of the new object types, you must define the object types by using an `com.ibm.mq.explorer.tests.ContextGroup` extension. This extension displays a new high-level group in the Run Tests dialog on the **Objects** page at the level of the supplied **Queue Managers**, **Clusters**, and **Queue sharing groups** groups.

To define a new object type:

Procedure

1. On the **Extensions** page of the `plugin.xml` file, add the `com.ibm.mq.explorer.tests.ContextGroup` extension to the **All Extensions** pane.
2. Configure the new group according to the details in the following table:

Attribute	Description	Example value
groupId	The unique identifier of the group that you are creating.	<code>com.ibm.mq.explorer.tests.samples.NewGroup</code>
name	A meaningful name for the group.	My New Group
description	A brief description of the group.	This is my first new group.

You have defined the new group. Next, define the criteria that is used to identify which group an object belongs to.

3. In the **All Extensions** pane, right-click the group, select **New**, and then select the type of criteria to use according to the information in the following table:

Criteria type	Description	Example value
instanceOf	The object must use an instance of a specific fully qualified class.	<code>com.ibm.mq.explorer.clusterplugin.internal.objects.ClusterObject</code>
objectType	The object's <code>objectType</code> attribute must have a specific value. You can also specify whether the value must match the criteria exactly.	<code>com.ibm.mq.explorer.queuemanager</code>
objectId	The object's <code>objectId</code> attribute must have a specific value. You can also specify whether the value must match the criteria exactly.	<code>com.ibm.mq.explorer.queuemanager</code>

4. Save the `plugin.xml` file.

Results

You have defined the new group of objects for which you can run tests.

Writing your own tests: Sample 1

The following source code is an example of a skeleton test that returns static data. The test is given here as an example of the WMQTest interface.

```
/*
 * Licensed Materials - Property of IBM
 *
 * 63H9336
 * (c) Copyright IBM Corp. 2005, 2022. All Rights Reserved.
 *
 * US Government Users Restricted Rights - Use, duplication or
 * disclosure restricted by GSA ADP Schedule Contract with
 * IBM Corp.
 */

package com.ibm.mq.explorer.tests.sample;

/**
 * Sample test that is run from an additional test in the WMQ standards test tree
 */
public class WMQTestSimple extends WMQTest {

    /*
     * (non-Javadoc)
     *
     * @see
     * com.ibm.mq.explorer.tests.WMQTest#runTest(com.ibm.mq.explorer.tests.internal.actions.WMQTestEngine,
     * org.eclipse.core.runtime.IProgressMonitor, com.ibm.mq.explorer.ui.extensions.MQExtObject[],
     * java.lang.String)
     */
    public void runTest(WMQTestEngine callback, IProgressMonitor guimonitor,
        MQExtObject[] contextObjects, TreeNode treenodeId) {

        // Start with the default implementation. this will store a handle
        // to the test engine that will be needed when we want to submit
        // any results at the end of the test
        super.runTest(callback, guimonitor, contextObjects, treenodeId);

        // prepare space to store test results
        ArrayList testresults = new ArrayList();

        // initialise the progress bar part of the GUI used to show progress (4 stages)
        guimonitor.beginTask(getTestName(), 4);

        // Loop through 4 times, incrementing the progress counter by 1 each time
        for (int k = 0; k < 4; k++) {
            try {
                // Sleep for a bit so it looks like we are doing some work
                Thread.sleep(900);
            }
            catch (InterruptedException e) {
            }

            // increment GUI progress bar used to show progress, completed 1 sleep
            guimonitor.worked(1);
        }

        // Create a new test result and add it to our array list of results
        testresults.add(new WMQTestResult(IMarker.SEVERITY_INFO, "SAMPLE: Our addition test
        worked!", //$NON-NLS-1$
            "Object name", getTestSubCategory())); //$NON-NLS-1$

        // package up results and return - test complete.
        testComplete((WMQTestResult[]) testresults.toArray(new WMQTestResult[testresults.size()]));
    }
}
```

Writing your own tests: Sample 2

The following source code is an example of a test that checks queue names against a defined naming convention. If any queues are found with names that do not meet the defined naming convention, the details are displayed in the Test Results view.

```
/*
 * Licensed Materials - Property of IBM
 *
 * 5724-H72, 5655-L82, 5724-L26, 5655R3600
 *
 * (c) Copyright IBM Corp. 2005, 2022
 *
 * US Government Users Restricted Rights - Use, duplication or
 * disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
 */
package com.ibm.mq.explorer.tests.sample;

/**
 * A sample test used to check Queue Names against naming conventions. Queue names are checked
 * if
 * they begin with any of a set range of prefixes, defined in this class. Any names which do not
 * start with one of the prefixes are output in an error.
 *
 * This example uses the PCF classes provide by the MS0B SupportPac. Download the SupportPac
 * from
 * the IBM website, then include the jar file in the build path for the project.
 */
public class WMQQueueNames extends WMQTest {

    /** Maintain a count of how many queue managers we are waiting for replies from. */
    private static int numberOfQmgrs = 0;

    /** Stores the accepted queue name prefixes. */
    private static final String[] ACCEPTED_Q_PREFIXES = {"SALES_", "MARKETING_", "SHIPPING_", //
        $NON-NLS-1$//$NON-NLS-2$ //$NON-NLS-3$
        "INCOMING_", "OUTGOING_"}; //$NON-NLS-1$//$NON-NLS-2$

    /** Stores the user preference for whether system queues should be included. */
    boolean includeSystemObjs = false;

    /**
     * Starts the test.
     *
     * @param callback handle to the test engine running the test
     * @param guimonitor a handle to the object monitoring the test, provided to allow the test to
     * periodically check if the user has tried to cancel the test running and provide additional
     * user
     * feedback
     * @param contextObjects context MQExtObjects passed to the test engine
     * @param treeNodeId the treeNodeId used to launch the tests
     */
    public void runTest(WMQTestEngine callback, IProgressMonitor guimonitor,
        MQExtObject[] contextObjects, TreeNode treeNodeId) {

        // start with the default implementation. this will store a handle
        // to the test engine that will be needed when we want to submit
        // any results at the end of the test
        super.runTest(callback, guimonitor, contextObjects, treeNodeId);

        // prepare space to store any results we might want to return
        ArrayList testResults = new ArrayList();

        // get from Preferences whether we should include system queues
        includeSystemObjs = PreferenceStoreManager.getIncludeSysObjsPreference();

        // get a list of queue managers from the Explorer
        ArrayList allQmgrs = new ArrayList();

        for (int k = 0; k < contextObjects.length; k++) {
            if (contextObjects[k] instanceof MQQmgrExtObject) {
                // Object is a queue manager, add to list
                allQmgrs.add(contextObjects[k]);
            }
        }

        // how many queue managers are there?
        numberOfQmgrs = allQmgrs.size();
    }
}
```

```

// use the number of queue managers as a guide to track progress
guimonitor.beginTask(getTestName(), numberOfQmgrs);

// for each queue manager, submit a query
for (int i = 0; i < numberOfQmgrs; i++) {

    // get next queue manager
    MQQmgrExtObject nextQueueManager = (MQQmgrExtObject) allQmgrs.get(i);

    // only submit queries to connected queue managers
    if (nextQueueManager.isConnected()) {

        // get the name of the queue manager, for use in GUI
        String qmgrName = nextQueueManager.getName();

        // get a handle to a Java object representing the queue manager
        MQQueueManager qmgr = nextQueueManager.getMQQueueManager();

        try {
            // get a PCF message agent to handle sending PCF inquiry to
            PCFMessageAgent agent = new PCFMessageAgent(qmgr);

            // use PCF to submit an 'inquire queue names' query
            PCFMessage response = submitQueueNamesQuery(qmgrName, agent);

            // did we get a response to the query?
            if (response != null) {
                // get the queue names out of the reply
                String[] qnames = (String[]) response.getParameterValue(CMQCFC.MQCACF_Q_NAMES);

                // check each name
                for (int j = 0; j < qnames.length; j++) {
                    boolean qnameOkay = checkQueueName(qnames[j]);

                    if (!qnameOkay) {
                        // if a problem was found with the name, we generate an
                        // error message, and add it to the collection to be
                        // returned
                        testResults.add(generateTestResult(qnames[j], qmgrName));
                    }
                }
            }
        } catch (MQException e) {
            // record error details
            e.printStackTrace();
        }
    }

    // finished examining a queue manager
    guimonitor.worked(1);
}

// return any results that this test has generated
WMQTestResult[] finalresults = (WMQTestResult[]) testResults
    .toArray(new WMQTestResult[testResults.size()]);
testComplete(finalresults);
}

/**
 * Used internally to submit a INQUIRE_Q_NAMES query using PCF to the given queue manager.
 *
 * @param qmgrName name of the queue manager to submit the query to
 * @param agent
 * @return the PCF response from the queue manager
 */
private PCFMessage submitQueueNamesQuery(String qmgrName, PCFMessageAgent agent) {

    // build the pcf message
    PCFMessage inquireQNames = new PCFMessage(CMQCFC.MQCMD_INQUIRE_Q_NAMES);
    inquireQNames.addParameter(CMQC.MQCA_Q_NAME, "*"); //$NON-NLS-1$

    try {
        // send the message
        PCFMessage[] responseMsgs = agent.send(inquireQNames);

        // check if results received successfully
        if (responseMsgs[0].getCompCode() == 0) {
            return responseMsgs[0];
        }
    }
}

```

```

    }
    catch (IOException e) {
        // record error details
        e.printStackTrace();
    }

    catch (MQException e) {
        // record error details
        e.printStackTrace();
    }

    // for some reason, we don't have a response, so return null
    return null;
}

/**
 * Used internally to check the given queue name against the collection of acceptable
 * prefixes.
 *
 * @param queueName queue name to check
 * @return true if the queue name is okay, false otherwise
 */
private boolean checkQueueName(String queueName) {

    // if this is a system object (i.e. it has a name which begins with
    // "SYSTEM.") we check the
    if ((queueName.startsWith("SYSTEM.")) || (queueName.startsWith("AMQ."))) { //$NON-NLS-1$//$NON-NLS-2$
        if (!includeSystemObjs) {
            // user has requested that we do not include system
            // objects in the test, so we return true to
            // avoid any problems being reported for this queue
            return true;
        }
    }

    // PCF response will white-pad the queue name, so we trim it now
    queueName = queueName.trim();

    // check the queue name against each of the acceptable prefixes
    // in turn, returning true immediately if it is
    for (int i = 0; i < ACCEPTED_Q_PREFIXES.length; i++) {
        if (queueName.startsWith(ACCEPTED_Q_PREFIXES[i]))
            return true;
    }

    // we have checked against all accepted prefixes, without
    // finding a match
    return false;
}

/**
 * Used internally to generate a test result for the given queue name.
 *
 * @param queueName queue name which doesn't meet requirements
 * @param qmgrName name of queue manager which hosts the queue
 * @return the generated test result
 */
private WMQTestResult generateTestResult(String queueName, String qmgrName) {
    String res = "Queue (" + queueName.trim() + ") does not begin with a known prefix"; //$NON-NLS-1$//$NON-NLS-2$

    return new WMQTestResult(IMarker.SEVERITY_ERROR, res, qmgrName, getTestSubCategory());
}
}

```

Writing your own tests: Sample 3

The following source code is an example of a test that shows an asynchronous approach to requesting data and processing data.

```

/*
 * Licensed Materials - Property of IBM
 *
 * 5724-H72, 5655-L82, 5724-L26, 5655R3600
 *
 * (c) Copyright IBM Corp. 2005, 2022

```

```

*
* US Government Users Restricted Rights - Use, duplication or
* disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*/
package com.ibm.mq.explorer.tests.sample;

/**
 * Pseudo-code sample demonstrating an asynchronous approach to implementing a
 * Test.
 */
public class QueuesTest extends WMQTest implements SomeListener {

    /** Used to store test results. */
    private ArrayList testresults = new ArrayList();

    /**
     * Used to start the test.
     * <p>
     * @param callback      handle to the test engine running the test
     * @param guimonitor    a handle to the object monitoring the test,
     *                      provided to allow the test to periodically check
     *                      if the user has tried to cancel the test running
     */
    public void runTest(WMQTestEngine callback, IProgressMonitor guimonitor, MQExtObject[]
contextObjects, TreeNode treeNodeId) {

        super.runTest(callback, guimonitor, contextObjects, treeNodeId);

        // reset all test stores
        testresults = new ArrayList();

        // initialise the progress bar part of the GUI used to show progress of
        // this test
        guimonitor.beginTask(getTestName(), numqmgrs);

        // start the test!

        // send query
        PseudoQueueManager qmgrHandle = pseudoGetQueueManager();
        submitQmgrQuery(qmgrHandle, this, query);

        // note that the runTest method is now finished, but the test is not
over!
    }

    /**
     * Used to process results received in response to the query submitted by
     * runTest.
     * <p>
     * @param objects      data received
     */
    public void dataReponseReceived(ArrayList objects) {

        // analyse each of the replies in the collection received in the reply
        for ( int i = 0; i < objects.size(); i++ ) {
            PseudoQueue nxtQueue = (PseudoQueue) objects.get(i);
            analyseQueue(nxtQueue);

            // increment GUI progress bar used to show progress of this test
            getGUIMonitor().worked(1);
        }

        // return the completed results
        WMQTestResult[] finalresults = (WMQTestResult[]) testresults.toArray(new
WMQTestResult[0]);
        testComplete(finalresults);
    }

    /**
     * Analyse the given queue. If any potential problems are found, a problem
     * marker is added to the testresults collection.
     * <p>
     * @param queue        queue to analyse
     */
    private void analyseQueue(PseudoQueue queue) {

        // do something

        // add a problem marker to the collection
    }
}

```

```

        if (problemFound) {
            testresults.add(new WMQTestResult(IMarker.SEVERITY_WARNING,
                "A problem was found with "
                + queueName,
                getQueueManagerName(queue),
                getTestSubCategory()));
        }
    }
}

```

Writing your own tests: Sample 4

The following source code is an example of a diagnostic tool. Use this code in place of real test code to write to the console the objects that will be accessed by the real test code.

```

/*
 * Licensed Materials - Property of IBM
 *
 * 63H9336
 * (c) Copyright IBM Corp. 2005, 2022. All Rights Reserved.
 *
 * US Government Users Restricted Rights - Use, duplication or
 * disclosure restricted by GSA ADP Schedule Contract with
 * IBM Corp.
 */
package com.ibm.mq.explorer.tests.sample;

/**
 * List all the context objects provided to standard out
 */
public class WMQTestSimple extends WMQTest {

    /*
     * (non-Javadoc)
     * @see
     * com.ibm.mq.explorer.tests.WMQTest#runTest(com.ibm.mq.explorer.tests.internal.actions.WMQTestEngine,
     * org.eclipse.core.runtime.IProgressMonitor, com.ibm.mq.explorer.ui.extensions.MQExtObject[],
     * java.lang.String)
     */
    public void runTest(WMQTestEngine callback, IProgressMonitor guimonitor,
        MQExtObject[] contextObjects, TreeNode treenodeId) {

        super.runTest(callback, guimonitor, contextObjects, treenodeId);

        // prepare space to store test results
        ArrayList testresults = new ArrayList();

        // Loop through all supplied MQExtObjects and output them to the console
        System.out.println("Objects supplied to this test:"); //$NON-NLS-1$
        for (int k = 0; k < contextObjects.length; k++) {
            if (contextObjects[k] != null) {
                System.out.println(contextObjects[k].getName());
            }
        }

        // Output the tree node ID to the console
        System.out.println("tree node ID supplied to this test: " + treenodeId); //$NON-NLS-1$

        // Add a test result
        testresults.add(new WMQTestResult(IMarker.SEVERITY_WARNING,
            "SAMPLE: Listing context completed", //$NON-NLS-1$
            "Object name", getTestSubCategory())); //$NON-NLS-1$

        // package up results and return - test complete.
        testComplete((WMQTestResult[]) testresults.toArray(new WMQTestResult[testresults.size()]));
    }
}

```


Sending test messages

You can use a test message to check whether an application or a queue manager can put a message on a queue. You can also browse messages that are already on a queue or clear messages from a queue.

About this task

You can verify whether an application or a queue manager can put a message on a queue by using IBM MQ Explorer to put a test message on the queue. For instructions, see [Putting a test message on a queue](#).

You can also use IBM MQ Explorer to browse messages that are already on a queue. Browsing a queue enables you to view the messages that are on the queue without getting (removing) them from the queue. For instructions, see [Browsing the messages on a queue](#).

Finally, you can use IBM MQ Explorer to clear messages from a queue without having to stop and restart the queue manager. For instructions, see [Clearing the messages from a queue](#).

Putting a test message on a queue

You can use a test message to verify whether an application or a queue manager can put a message on a queue.


About this task

To put a test message on a queue, complete the following steps.

Procedure

1. In the Navigator view, click the **Queues** folder that contains the queue. The queue is displayed in the Content view.
2. In the Content view, right-click the queue, then click **Put Test Message...** The Put Test Message dialog opens.
3. In the **Message data** field, type some sample message data. For example, type `This is a test message`.
4. Click **Put Message**. The message is put on the queue.
5. Click **Close** to close the Put Test Message dialog.

Results

In the Content view, the value in the **Current queue depth** column for the queue is incremented by one. If the value has not changed, click Refresh  on the Content view toolbar.

Related tasks

[“Sending test messages” on page 89](#)

You can use a test message to check whether an application or a queue manager can put a message on a queue. You can also browse messages that are already on a queue or clear messages from a queue.

[“Browsing the messages on a queue” on page 89](#)

Browsing a queue enables you to view the messages that are on the queue without getting (removing) them from the queue.

[“Clearing the messages from a queue” on page 90](#)

You can clear messages from a queue without having to stop and restart the queue manager.

Browsing the messages on a queue

Browsing a queue enables you to view the messages that are on the queue without getting (removing) them from the queue.

About this task

To browse the messages on a queue, complete the following steps.

Procedure

1. In the Navigator view, click the **Queues** folder that contains the queue.
The queue is displayed in the Content view.
2. In the Content view, right-click the queue, then click **Browse Messages...**
The Message Browser dialog opens.

Results

The **Message browser** window displays a user-defined number of bytes from a user-defined number of messages, with the most recent message at the end of the list. Double-click a message to view its properties, including the data in the message. All of the messages remain on the queue.

Set the number of messages and number of bytes to be displayed in the **Preferences** window as described in [“Configuring IBM MQ Explorer” on page 223](#).

Related tasks

[“Sending test messages” on page 89](#)

You can use a test message to check whether an application or a queue manager can put a message on a queue. You can also browse messages that are already on a queue or clear messages from a queue.

[“Putting a test message on a queue” on page 89](#)

You can use a test message to verify whether an application or a queue manager can put a message on a queue.

[“Clearing the messages from a queue” on page 90](#)

You can clear messages from a queue without having to stop and restart the queue manager.

Clearing the messages from a queue

You can clear messages from a queue without having to stop and restart the queue manager.

About this task

To clear all the messages from a queue, complete the following steps.

Procedure

1. In the Navigator view, click the **Queues** folder that contains the queue.
The queue is displayed in the Content view.
2. In the Content view, right-click the queue, then click **Clear Messages...**
The Clear Queue dialog opens.
3. Select the method to use to clear the messages from the queue:
 - If you use the CLEAR command, all of the messages are cleared from the queue. However, if the queue is already opened exclusively by another application or if the queue contains uncommitted messages, the command fails immediately and none of the messages are cleared.
 - If you use the MQGET API call, the messages are got from the queue until no more messages are available. However, MQGET does not recognize uncommitted messages, which means that there could still be uncommitted messages on the queue. Also, the command might fail if the queue is already exclusively opened by another application.
4. Click **Clear**.
A message is displayed to tell you whether the command was successful.
5. Click **Close** to close the dialog.

Results

All the messages are cleared from the queue unless there was a problem; for example, the queue contains uncommitted messages.

Related tasks

[“Sending test messages” on page 89](#)

You can use a test message to check whether an application or a queue manager can put a message on a queue. You can also browse messages that are already on a queue or clear messages from a queue.

[“Putting a test message on a queue” on page 89](#)

You can use a test message to verify whether an application or a queue manager can put a message on a queue.

[“Browsing the messages on a queue” on page 89](#)

Browsing a queue enables you to view the messages that are on the queue without getting (removing) them from the queue.

Starting and stopping objects and services

Before you can create objects for a queue manager, the queue manager must be running. Similarly, before an application can send messages through a channel, the channel must be running and the receiving queue manager must have a running listener. In addition, any services like channel initiators and trigger monitors must be running if they are needed.

About this task

For more information, see the following topics:

- [Starting and stopping a queue manager](#)
- [Starting and stopping a channel](#)
- [Starting and stopping a listener](#)
- [Starting and stopping a command server](#)
- [Starting and stopping a custom service](#)
- [Starting a trigger monitor](#)
- [Starting a channel initiator](#)

Starting and stopping a queue manager

You can start or stop an individual queue manager, or start or stop all the queue managers in a queue manager set.

About this task

Before you can create IBM MQ objects to be hosted by the queue manager, and before you can start any of the IBM MQ objects that are hosted by the queue manager, you must start the queue manager.

In some circumstances, such as if you have changed the attributes of the queue manager, if you want to apply a fix pack to IBM MQ, or if you want to stop the queue manager participating in a messaging network, you need to stop the queue manager.

To start or stop a queue manager in IBM MQ Explorer, complete the following steps.

Procedure

1. In the Navigator view, expand the **Queue Managers** folder.
2. Right-click the name of the queue manager, then click **Start** or **Stop**.
3. If you chose to stop the queue manager, select **Controlled** or **Immediate**.
4. Click OK.

Results

The icon next to the queue manager's name changes to indicate that the queue manager has started or stopped as appropriate.

Queue manager sets

Before you begin

You can also start or stop all the queue managers in a queue manager set.

Before you start or stop all the queue managers in a set, the following steps need to be taken:

1. You must display queue manager sets as described in: [“Displaying queue manager sets” on page 233](#).
2. You must define a set for the queue managers as described in: [“Defining manual sets” on page 234](#) or: [“Defining automatic sets” on page 235](#).

About this task

To start or stop all the queue managers in a set in the **Explorer** view:

Procedure

1. In the Navigator view, expand the **Queue Managers** folder.
2. Right-click the name of the set to open the menu. Click **Start Local Queue Managers** or **Stop Local Queue Managers**.

Results

The icon next to the queue manager's name changes to indicate that the queue manager has started or stopped as appropriate.

Related concepts

[“Queue managers” on page 28](#)

A queue manager is a program that provides messaging services to applications. Applications that use the Message Queue Interface (MQI) can put messages on queues and get messages from queues. The queue manager ensures that messages are sent to the correct queue or are routed to another queue manager.

[“Objects in IBM MQ Explorer” on page 27](#)

In IBM MQ Explorer, all of the queue managers and their IBM MQ objects are organized in folders in the Navigator view.

Reconnectable clients

IBM MQ clients can take advantage of automatic reconnection if their connection to a queue manager is broken. This is of value when a connection breaks, or a queue manager fails. When you stop a queue manager you have the option of enabling the automatic reconnection of clients.

There are a number of ways to code and configure an IBM MQ MQI client to make it continue to work if the queue manager to which it is connected fails. An application program can respond to a queue manager failure by closing queues and subscriptions and disconnecting from the failing queue manager. The client program might then attempt to reconnect, and either wait until the queue manager is running again, or connect to another queue manager in the same queue manager group.

To make this common procedure easier, a client program can connect to a queue manager with the option of being automatically reconnected to another queue manager (or reconnected to this queue manager) if the current connection fails. No application programming is required. The application program does not have to be notified of any broken connection errors from the queue manager.

Automatic client reconnection is not supported by IBM MQ classes for Java.

As the IBM MQ administrator, you might want to signal to all client application programs, including ones that have requested queue manager failures to be handled automatically, that you are stopping

the queue manager deliberately, and want client applications to stop, rather than have the client applications treat the queue manager stoppage as a failure and attempt to reconnect automatically. This is the default behavior of the **Stop queue manager** command, to maintain compatibility with earlier releases of IBM MQ. However, as an option on the Stop queue manager command, you can use the `Instruct reconnectable clients to reconnect` option, and the indication that the queue manager is stopping is intercepted by a reconnectable client connection, and it starts trying to reconnect automatically as if a failure had occurred.

Related information

[Automatic client reconnection](#)

Starting and stopping a channel

The way in which a channel is started depends on whether it is a caller channel or a responder channel. When you stop a channel, you can choose whether to stop the channel after the current batch of messages has finished processing, or force the channel to shut down before the current message batch has finished processing.

About this task

Channels can be categorized as either *callers* or *responders*. An application starts caller channels either directly, or automatically by using a channel initiator. Responder channels can be started only by the listener.

Note that you must have the required transport protocol, for example, TCP/IP, on the computers at each end of the channel.

Starting a channel manually

About this task

You can start caller channels in IBM MQ Explorer. When you start responder channels in IBM MQ Explorer, you are actually changing the responder channel's state from Stopped to Inactive; the listener then changes the state from Inactive to Started. You must, therefore, start a listener on your computer if you are using responder channels.

To start a channel, complete the following steps.

Procedure

1. In the Navigator view, click the **Channels** folder to display the channels in the Content view.
2. In the Content view, right-click the channel, then click **Start**.

Results

The channel starts. The icon next to the channel changes to show that the channel is running.

Stopping a channel

About this task

To stop a channel, complete the following steps.

Procedure

1. In the Navigator view, click the **Channels** folder to display the channels in the Content view.
2. In the Content view, right-click the channel, then click **Stop....**
The Stop Channel dialog opens.
3. Select how IBM MQ stops the channel:

- Accept the default values (do not select the check boxes) to end the channel after the current batch of messages has finished processing (on Windows, Linux, UNIX, or IBM i), or to end the channel after the current message (on z/OS). For a receiving channel, if there is no batch in progress, the channel waits for either the next batch or the next heartbeat (if heartbeats are being used) before stopping. For server-connection channels, the channel stops when the connection ends.
 - Select the **Force interruption of current message batch** check box to terminate the transmission of any current batch; the channel's thread or process is not terminated. This is likely to result in in-doubt channels. For server-connection channels, the current connection is broken.
 - Select the **Allow process/thread termination** check box if you select the **Force interruption of current message batch** check box and you want to terminate the channel thread or process.
4. If the channel definition is a responder channel, multiple queue managers or remote connections can be using the same responder channel. You can, therefore, filter which channels are stopped: select the relevant check box then type the name of the queue manager or remote connection.
 5. Select the state that the channel will change to when it stops:
 - Click **Stopped** to stop the channel but keep the process or thread running; the channel is still active and consuming resources.
 - Click **Inactive** to stop the channel, including stopping the process or thread; the channel is inactive and is not consuming resources.

Results

The channel stops running. The icon next to the channel changes to show that the channel is no longer running.

Related concepts

[“Listeners” on page 38](#)

A listener is an IBM MQ process that listens for connections to the queue manager.

[“Channel initiators” on page 45](#)

A channel initiator is an application that processes the trigger messages that are put on initiation queues when a trigger event occurs. A channel initiator is a special type of trigger monitor that starts channels rather than applications.

[“Channels” on page 34](#)

IBM MQ can use three different types of channels: a message channel, an MQI channel, and an AMQP channel.

Starting and stopping a listener

Each listener object in IBM MQ Explorer represents a listener process; when you start the listener object in IBM MQ Explorer, the listener process starts.

About this task

For a queue manager to receive messages from channels, it must have a running listener that is correctly configured for its transport type. The listener will then start the receiving end of the channel when it detects that an application has started the sending end of the channel.

To start or stop a listener, complete the following steps.

Procedure

1. In the **Navigator** view, click the **Listeners** folder to display the listeners in the **Content** view.
2. In the **Content** view, right-click the listener, then click **Start** or **Stop**.

Results

The listener starts or stops as appropriate.



Listeners on the z/OS platform are not listener objects and do not behave in the same way as listener objects. When listeners on the z/OS platform are stopped, they are no longer associated with the z/OS queue manager.

Related concepts

[“Listeners” on page 38](#)

A listener is an IBM MQ process that listens for connections to the queue manager.

[“Channels” on page 34](#)

IBM MQ can use three different types of channels: a message channel, an MQI channel, and an AMQP channel.

Related tasks

[“Starting and stopping a channel” on page 93](#)

The way in which a channel is started depends on whether it is a caller channel or a responder channel. When you stop a channel, you can choose whether to stop the channel after the current batch of messages has finished processing, or force the channel to shut down before the current message batch has finished processing.

[“Creating and configuring queue managers and objects” on page 27](#)

You can create, configure, and delete queue managers and objects in IBM MQ Explorer by using the Navigator view and Content view.

Starting and stopping a command server

To connect to a queue manager from IBM MQ Explorer, the queue manager command server must be running.

About this task

To start or stop the command server, complete the following steps.

Procedure

In the Navigator view, right-click the queue manager, then click **Start Command Server** or **Stop Command Server**.

Results

The command server starts or stops as appropriate.

Related tasks

[“Starting and stopping a queue manager” on page 91](#)

You can start or stop an individual queue manager, or start or stop all the queue managers in a queue manager set.

Starting and stopping a custom service

You can configure a custom service to start automatically when the queue manager starts. You can also start or stop a service manually.

About this task

You can configure a custom service to automatically start when the queue manager starts by changing the value of the `Service control` attribute in the service's properties dialog. The following instructions describe how to manually start a service.

Note that if there is no end command defined in the service, for example, for trigger monitors, when the service stops, the object that the service is controlling does not stop.

To start or stop a service, complete the following steps.

Procedure

1. In the Navigator view, click the **Services** folder to display the services in the Content view.
2. In the Content view, right-click the service, then click **Start** or **Stop**.

Results

The service starts or stops as appropriate. The icon next to the service changes to show whether the service is running.

Related concepts

[“Custom services” on page 45](#)

Custom services are services that you create to run commands automatically.

[“Trigger monitors” on page 44](#)

A trigger monitor is an application that processes the trigger messages that are put on initiation queues when a trigger event occurs.

Starting a trigger monitor

To start a trigger monitor, you must first create a service that will start the trigger monitor.

About this task

To start a trigger monitor from IBM MQ Explorer, you must first create a service that will run the `runmqtrm` command (to start the trigger monitor) when the service starts.

Note that if you are starting a trigger monitor for a client, use the `runmqtrmc` command instead. For more information about trigger monitors, see [Trigger monitors](#) in the IBM MQ online product documentation.

To start a trigger monitor, complete the following steps.

Note: When the trigger monitor runs as a service, the started process or application runs in the background.

Procedure

1. In the Navigator view, expand the queue manager on which you want to start the trigger monitor service.
2. Right-click the queue manager's **Services** folder, then click **New... > Service**. The New Service dialog opens.
3. In the New Service dialog, type a name for the service, for example, `TriggerMonitor`, then click **Next**. You can now configure the new service.
4. Optional: In the **Description** field, type a description of the service, for example, `A trigger monitor for queue manager QM1`.
5. In the **Service control** field, configure how the service starts and stops:
 - To start and stop the service automatically when the queue manager starts and stops, click **Queue Manager**
 - To start the service automatically when the queue manager starts but not to stop when the queue manager stops, click **Queue Manager Start**
 - To configure the service so that you must manually start and stop it, click **Manual**.
6. In the **Start Command** field, type the full path to the `runmqtrm` command.
 - Type: `MQ_INSTALLATION_PATH\bin\runmqtrm` where `MQ_INSTALLATION_PATH` is replaced by the high-level directory in which IBM MQ is installed.
7. If the queue manager is not the default queue manager, in the **Start args** field, type `-m queue_manager_name` where `queue_manager_name` is the name of the queue manager.

8. If you want to use a queue other than `SYSTEM.DEFAULT.INITATION.QUEUE` as the initiation queue, in the **Start args** field, type `-q initq_name` where *initq_name* is the name of the queue.
9. In the **Service Type** field, select the type of service to run:
 - If you select **Command**, you can run multiple instances of the service but you cannot view the status of the service in IBM MQ Explorer.
 - If you select **Server**, you can run only one instance of the service but you can view the status of the service in IBM MQ Explorer.
10. Click **Finish**.

The new service is created on the selected queue manager.
11. Start the service.

For instructions, see [“Starting and stopping a custom service” on page 95](#).

Results

The service starts and runs the `runmqtrm` command, which starts the trigger monitor on the queue manager.

When you have started a trigger monitor, it just continues monitoring the specified initiation queue. You cannot stop a trigger monitor directly. When you stop the trigger monitor's queue manager, the trigger monitor stops too.

Related concepts

[“Trigger monitors” on page 44](#)

A trigger monitor is an application that processes the trigger messages that are put on initiation queues when a trigger event occurs.

Starting a channel initiator

To start a channel initiator, you must first create a service that will start the channel initiator.

About this task

Because a channel initiator is just a special type of trigger monitor, to start a channel initiator from IBM MQ Explorer, you must first create a service that will run the `runmqchl` command (to start the channel initiator) when the service starts.

The following instructions assume that you are creating a service called `ChannelInitiator` on a queue manager called `QM1`. For more information about channel initiators, see [Starting and stopping the channel initiator](#) in the IBM MQ online product documentation.

To create the channel initiator service:

Procedure

1. In the Navigator view, expand the queue manager, `QM1`, that you want to start the channel initiator on.
2. Right-click the queue manager's **Services** folder, then click **New... > Service**. The New Service dialog opens.
3. In the New Service dialog, type a name for the service, for example, `ChannelInitiator`, then click **Next**. You can now configure the new service, `ChannelInitiator`.
4. Optional: In the **Description** field, type a description of the `ChannelInitiator` service, for example, A channel initiator for queue manager `QM1`.
5. In the **Service control** field, configure how the service starts and stops:
 - To start and stop the service automatically when the queue manager starts and stops, click **Queue Manager**

- To start the service automatically when the queue manager starts but not to stop when the queue manager stops, click **Queue Manager Start**
 - To configure the service so that you must manually start and stop it, click **Manual**.
6. In the **Start Command** field, type the full path to the `runmqchi` command.
 - Type: `MQ_INSTALLATION_PATH\bin\runmqchi` where `MQ_INSTALLATION_PATH` is replaced by the high-level directory in which IBM MQ is installed.
 7. If QM1 is not the default queue manager, in the **Start args** field, type `-m QM1`
 8. If you want to use a queue other than `SYSTEM.CHANNEL.INITQ` as the initiation queue, in the **Start args** field, type `-q initq_name` where `initq_name` is the name of the queue.
 9. In the **Service Type** field, select **Command**.
 10. Click **Finish**.

The new service, ChannelInitiator, is created on the selected queue manager, QM1.
 11. Start the service.

For instructions, see [“Starting and stopping a custom service” on page 95](#).

Results

The service, ChannelInitiator, starts and runs the `runmqchi` command, which starts the channel initiator on the queue manager, QM1.

Related concepts

[“Trigger monitors” on page 44](#)

A trigger monitor is an application that processes the trigger messages that are put on initiation queues when a trigger event occurs.

[“Channel initiators” on page 45](#)

A channel initiator is an application that processes the trigger messages that are put on initiation queues when a trigger event occurs. A channel initiator is a special type of trigger monitor that starts channels rather than applications.

Showing or hiding a queue manager

By default, the Navigator view shows all of the queue managers on the computer on which IBM MQ Explorer is installed. However, if you have any queue managers that you are not currently administering, you can, if you wish, choose to hide them. You can also show and hide remote queue managers.

About this task

Before you can administer a queue manager in IBM MQ Explorer, you must show the queue manager in the **Queue Managers** folder in the IBM MQ Explorer Navigator view. By default, all of the queue managers on the computer on which IBM MQ Explorer is installed are detected automatically and are shown in the **Queue Managers** folder. You can hide queue managers if you don't want to administer them in IBM MQ Explorer.

If you are configuring JMS administered objects, you can add a queue manager to IBM MQ Explorer from a connection factory that defines the queue manager's connection details. You are actually creating a connection to IBM MQ Explorer from the connection factory that defines the queue manager's details.

You can also administer remote queue managers using cluster connections if you are already connected to a queue manager that belongs to a cluster to which the remote queue manager also belongs.

The following topics describe how to show and hide local and remote queue managers in IBM MQ Explorer:

- [Show local queue managers](#)
- [Show remote queue managers](#)
- [Hide queue managers](#)

- [Show hidden queue managers](#)
- [Remove queue managers](#)
- [“Adding a queue manager from a JMS connection factory” on page 56](#)
- [Administering a remote cluster queue manager](#)

Related concepts

[“Queue manager clusters” on page 48](#)

A cluster is a group of two or more queue managers that are logically associated and can share information with each other. Any queue manager can send a message to any other queue manager in the same cluster without you needing to set up a specific channel definition, remote queue definition, or transmission queue, because all of this information is held in the repository, to which all queue managers in the cluster have access.

Related tasks

[“Administering remote queue managers” on page 111](#)

In IBM MQ Explorer, you can enable IBM MQ queue managers on a remote computer for remote administration.

Showing a local queue manager

If you want to administer a local queue manager, it must appear in the Navigator view.

About this task

All of the queue managers that are hosted on the computer on which IBM MQ Explorer is installed are automatically detected and shown in the **Queue Managers** folder in IBM MQ Explorer but you can hide them from view. To administer a local queue manager using IBM MQ Explorer, you must show the queue manager in the **Queue Managers** folder in the Navigator view of IBM MQ Explorer.

If you have hidden the queue manager from view, you can show it again. For instructions, see: [Showing hidden queue managers](#).

Related tasks

[“Hiding queue managers” on page 106](#)

You can hide from view any queue manager that is displayed in the Navigator view. If you hide a queue manager that is a member of one or more queue manager Sets, the queue manager is not displayed in any of those Sets.

[“Showing a remote queue manager” on page 99](#)

If you want to administer a remote queue manager, you must connect IBM MQ Explorer to the remote queue manager, so that the queue manager then appears in the Navigator view.

[“Administering remote queue managers” on page 111](#)

In IBM MQ Explorer, you can enable IBM MQ queue managers on a remote computer for remote administration.

[“Removing a queue manager” on page 108](#)

You can remove a queue manager from IBM MQ Explorer if you no longer want to administer it in IBM MQ Explorer.

Showing a remote queue manager

If you want to administer a remote queue manager, you must connect IBM MQ Explorer to the remote queue manager, so that the queue manager then appears in the Navigator view.

About this task

IBM MQ Explorer automatically discovers all the queue managers on the computer on which IBM MQ Explorer is installed. However, IBM MQ Explorer does not automatically discover queue managers on other computers.

To administer remote queue managers, you must manually connect IBM MQ Explorer to the remote queue manager and show the queue manager in the **Queue Managers** folder in IBM MQ Explorer.

Use one of the following methods to connect to a remote queue manager:

- Create a connection manually. Create a connection to the remote queue manager using the **Add Queue Manager** wizard. You can use the default SYSTEM.ADMIN.SVRCONN channel or a server-connection channel that you specify yourself.
- Create a connection using a client channel definition table. If you use a client channel definition table to configure the channel, you can for example, define security exits on the channel.
- Creating a new security-enabled connection. Create a new security-enabled connection to the remote queue manager.
- Connect using an existing connection. Connect to a remote queue manager using an existing connection that has been made by another queue manager.

You can also show remote cluster queue managers in the **Queue Managers** folder so that you can administer them from IBM MQ Explorer. For more information, see [Administering remote cluster queue managers](#).

If IBM MQ Explorer cannot connect to the remote queue manager for any reason (for example, the remote queue manager is not running), a dialog is displayed asking if you want to add the queue manager anyway. Click **Yes** and the queue manager is displayed in the **Queue Managers** folder but none of its details are available until it connects.

IBM MQ Explorer cannot connect to queue managers that are running on IBM MQ platforms that do not support remote administration. For more information about which IBM MQ platforms are supported, see [Administering remote queue managers](#).

Automatic client reconnect is not supported by IBM MQ classes for Java.

For more detailed information about CCDTs, see [Client channel definition table](#) in the IBM MQ online product documentation.

Creating a connection manually

About this task

Before you can create the connection, you must know the following information about the remote queue manager:

- The name of the queue manager.
- The name of the computer that hosts the queue manager.
- The port number of the queue manager's listener.
- The name of the server-connection channel on the queue manager that IBM MQ Explorer uses to connect to the queue manager. If you have enabled the queue manager for remote administration, the SYSTEM.ADMIN.SVRCONN channel is available. Otherwise, use SYSTEM.DEF.SVRCONN, a client channel definition table, or a server-connection channel that you have created and named.

To manually create a connection from IBM MQ Explorer to a remote queue manager, complete the following steps.

Procedure

1. Right-click on Queue Managers in the Navigator view, then click **Add Remote Queue Manager**. The **Add Queue Manager** wizard opens enabling you to create a connection.
2. In the **Queue manager name** field, type the name of the queue manager to which you want to connect.
3. Ensure that **Connect directly** is selected, then click **Next**.
4. Ensure that **Specify connection details** is selected, then type the following details:

- In the **Host name or IP address** field, type the name of the computer that hosts the remote queue manager; Use one of the following formats:
 - The short host name, for example, joho The remote computer must be in the same domain as your local computer.
 - The fully qualified host name, for example, joho.example.com Use this if the remote computer is in a different domain to your local computer.
 - The IP address, for example 127.0.0.1
- In the **Port number** field, type the port number; for example, 1416
- In the **Server-connection channel** field, type the name of the channel to use

To change the defaults that are used, see [“Specifying the default values used to connect to remote queue managers”](#) on page 256

5. Optional: Select the **Autoreconnect** check box to configure IBM MQ Explorer to automatically reconnect to the queue manager if the connection is lost.
6. Optional: Change the frequency with which IBM MQ Explorer refreshes its information about the queue manager. To prevent IBM MQ Explorer automatically refreshing its information about the queue manager, click **No queue manager refresh interval**; to specify a different refresh interval, click **Specify queue manager refresh interval**, then type the number of seconds that you want IBM MQ Explorer to wait before refreshing its information about the queue manager.
7. Click **Finish**.

Results

IBM MQ Explorer connects to the remote queue manager, and the queue manager is shown in the **Queue Managers** folder in the Navigator view.

Creating a connection using a client channel definition table

Instead of manually specifying the connection details of the remote queue manager, you can use a predefined client channel definition table. Using this method of connection means that you can, for example, configure the channel to use security exits.

Before you begin

Before you can create a connection using a client channel definition table, you must create the client channel definition table on the computer that hosts the remote queue manager, then copy the client channel definition table to the local computer (from which you want to connect to the remote queue manager).

About this task

IBM MQ Explorer connects to the remote queue manager using the client channel definition table, and the queue manager is shown in the **Queue Managers** folder in the Navigator view.

To connect to the remote queue manager using the client channel definition tables, complete the following tasks in IBM MQ Explorer on the local computer (the computer from which you want to connect to the remote queue manager):

Procedure

1. Right-click on Queue Managers in the Navigator view, then click **Add Remote Queue Manager**
The **Add Queue Manager** wizard opens enabling you to create a connection.
2. In the **Queue manager name** field, type the name of the queue manager to which you want to connect.
3. Ensure that **Connect directly** is selected, then click **Next**.
4. Click **Use client channel definition table**, then browse for the client channel definition table file.

5. Optional: Select the **Autoreconnect** check box to configure IBM MQ Explorer to automatically reconnect to the queue manager if the connection is lost.
6. Optional: Change the frequency with which IBM MQ Explorer refreshes its information about the queue manager. To prevent IBM MQ Explorer automatically refreshing its information about the queue manager, click **No queue manager refresh interval**; to specify a different refresh interval, click **Specify queue manager refresh interval**, then type the number of seconds that you want IBM MQ Explorer to wait before refreshing its information about the queue manager.
7. Click **Finish**.

Results

Creating a security-enabled connection

About this task

For more information about using TLS with client connections, see [Secure Sockets Layer \(SSL\) support in IBM MQ classes for Java](#) in the IBM MQ online product documentation.

To connect to a remote queue manager using a security-enabled connection, complete the following tasks in IBM MQ Explorer on the computer from which you want to connect to the remote queue manager.

Note: If you place the IBM MQ Explorer plug-ins in another Eclipse environment, to use the full set of CipherSuites, and to operate with certified FIPS 140-2 or Suite-B compliance, a suitable JRE is required. IBM Java 7 Service Refresh 4, Fix Pack 2 or a higher level of the IBM JRE provides the appropriate support.

Procedure

1. Right-click on Queue Managers in the Navigator view, then click **Add Remote Queue Manager**. The **Add Queue Manager** wizard opens, enabling you to create a connection.
2. In the **Queue manager name** field, type the name of the queue manager to which you want to connect.
3. Ensure that **Connect directly** is selected, then click **Next**.
4. Ensure that **Specify connection details** is selected, then type the following details:
 - In the **Host name or IP address** field, type the name of the computer that hosts the remote queue manager; Use one of the following formats:
 - The short host name, for example, `joho`. The remote computer must be in the same domain as your local computer.
 - The fully qualified host name, for example, `joho.example.com`. Use this if the remote computer is in a different domain to your local computer.
 - The IP address, for example `127.0.0.1`.
 - In the **Port number** field, type the port number; for example, `1416`.
 - In the **Server-connection channel** field, type the name of the channel to use.

To change the defaults that are used, see [“Specifying the default values used to connect to remote queue managers” on page 256](#).

5. Optional: Select the **Autoreconnect** check box to configure IBM MQ Explorer to automatically reconnect to the queue manager if the connection is lost.
6. Optional: Change the frequency with which IBM MQ Explorer refreshes its information about the queue manager. To prevent IBM MQ Explorer automatically refreshing its information about the queue manager, click **No queue manager refresh interval**; to specify a different refresh interval, click **Specify queue manager refresh interval**, then type the number of seconds that you want IBM MQ Explorer to wait before refreshing its information about the queue manager.
7. Click **Next**.

Results

At this point in the wizard, you can select the optional security parameters on the new pages of the wizard. All the security parameters are optional and you are not required to enable any of them if you do not want to, however, you must select **Enable SSL stores** to access the **Enable SSL options** parameters:

1. Optional. Select **Enable security exit** and type your security exit details into the fields. The remote server conn channel must also have a security exit defined. Click **Next**.
2. Optional. Select **Enable user identification** and type your required user identification details in the field. If you want to set the optional password, type your password details in the field. Optional: The remote server conn channel can also have a security exit defined. Click **Next**.
3. Optional. Select **Enable SSL stores** to specify TLS certificate key repository details. The remote server conn channel must also have TLS enabled. To specify certificate stores choose either one or both of the following options.
 - Optional. Click **Browse** in the **Selected Certificate Store** section of the dialog to locate the certificate store file. If you want to set the optional password, click **Enter password...** to open the **Password details** dialog where you must type your password details in the fields.
 - Optional. Click **Browse** in the **Personal certificate Store** section of the dialog to locate your personal certificate store file. You must set a password when defining a personal certificate store; click **Enter password...** to open the **Password details** dialog where you must type your password details in the fields.

Click **Next**.

4. Optional. Select **Enable SSL options**. Select the TLS options you require, and click **Finish** to create the TLS-enabled connection and close the wizard. You must have previously selected **Enable SSL stores** to access the **Enable SSL options** parameters.

Passwords used by the IBM MQ Explorer to connect to resources, for example, opening TLS stores or connecting to queue managers, can be stored in a file. The location of the file can be changed to a remote or removable device. For more information, see: [“Passwords preferences” on page 194](#).

IBM MQ Explorer now connects to the remote queue manager using a TLS-secured connection, and the queue manager is shown in the Queue Managers folder in the Navigator view.

Using an existing connection

About this task

IBM MQ Explorer connects to the remote queue manager and the queue manager is shown in the **Queue Managers** folder in the Navigator view.

You can also use existing cluster connections to administer remote cluster queue managers. For more information, see [Administering remote cluster queue managers](#).

To connect using an existing connection that has been made by another queue manager, complete the following steps.

Procedure

1. Right-click on Queue Managers in the Navigator view, then click **Add Remote Queue Manager**.
The **Add Queue Manager** wizard opens enabling you to create a connection.
2. In the **Queue manager name** field, type the name of the queue manager to which you want to connect.
3. Click **Connect using an intermediate queue manager**, then click **Next**.
4. From the **Intermediate queue manager** list, click the name of the queue manager that made the existing connection.
5. Optional: Select the **Autoreconnect** check box to configure IBM MQ Explorer to automatically reconnect to the queue manager if the connection is lost.

6. Optional: Change the frequency with which IBM MQ Explorer refreshes its information about the queue manager. To prevent IBM MQ Explorer automatically refreshing its information about the queue manager, click **No queue manager refresh interval**; to specify a different refresh interval, click **Specify queue manager refresh interval**, then type the number of seconds that you want IBM MQ Explorer to wait before refreshing its information about the queue manager.
7. Click **Finish**.

Related tasks

[“Administering remote queue managers” on page 111](#)

In IBM MQ Explorer, you can enable IBM MQ queue managers on a remote computer for remote administration.

[“Administering a remote cluster queue manager” on page 154](#)

After connecting to a remote cluster queue manager using the cluster information source as an intermediate queue manager, you can select to show the queue manager in the **Queue Managers** folder. You can use then use the connection to administer the remote queue manager.

[“Showing or hiding a queue manager” on page 98](#)

By default, the Navigator view shows all of the queue managers on the computer on which IBM MQ Explorer is installed. However, if you have any queue managers that you are not currently administering, you can, if you wish, choose to hide them. You can also show and hide remote queue managers.

Related reference

[“Passwords preferences” on page 194](#)

You can store passwords to a file so that you do not have to enter them every time you want to connect to resources.

Creating a client channel definition table

You can create a client channel definition table for a queue manager to make it easier to connect instances of IBM MQ Explorer to the queue manager.

About this task

When you connect IBM MQ Explorer to the queue manager using the client channel definition table, the table provides all of the connection information and you do not need to know the connection details to connect to the queue manager.

The following instructions describe how to create a client channel definition table that can be used for connections secured by Transport Layer Security (TLS). To create a client channel definition table that does not use TLS, just skip the steps about configuring TLS.

To create a client channel definition table, perform the following tasks on the computer that hosts the remote queue manager:

Procedure

1. If you want to secure connections that use the client channel definition table, configure the queue manager for using TLS-enabled connections.
2. Create a server-connection channel on the queue manager.
3. If you are using TLS, configure the server-connection channel to use TLS.
4. Create a client-connection channel, with the same name as the server-connection channel, on the queue manager.
5. If you are using TLS, configure the client-connection channel to use TLS.
If you have configured the server-connection channel to use TLS, you must also configure the client-connection channel to match.
6. Move the queue manager's client channel definition table to the computer from which you want to connect to the queue manager (the computer on which IBM MQ Explorer is installed). For example, use FTP to transfer the file between the two computers.

Results

Your new client channel definition table is now available for IBM MQ Explorer to use to connect to the remote queue manager.

Related tasks

[“Configuring TLS channels” on page 161](#)

To configure TLS channels, you use the **SSL** page of the **Channel properties** dialog to define the cipher specification to be used. You can optionally configure a channel to accept only certificates with attributes in the distinguished name of the owner that match given values. You can also optionally configure a queue manager channel so that the queue manager refuses the connection if the initiating party does not send its own personal certificate.

[“Creating and configuring queue managers and objects” on page 27](#)

You can create, configure, and delete queue managers and objects in IBM MQ Explorer by using the Navigator view and Content view.

Specifying the default location and default password of TLS certificates

You can configure IBM MQ Explorer to use TLS certificates in the TrustStore and KeyStore to connect to remote queue managers with an TLS-enabled connection.

About this task

To configure IBM MQ Explorer with the location and password of the TLS certificate store, complete the following tasks in IBM MQ Explorer on the computer from which you want to connect to the remote queue manager:

Procedure

1. In IBM MQ Explorer, click **Window > Preferences**.
The **Preferences** dialog opens.
2. Expand **MQ Explorer**.
3. Expand **Client Connections**. The default security settings dialogs are now accessible.
4. Select **SSL Key Repositories** to display the **SSL Key Repositories** pane.
5. In the **Trusted Certificate Store** field, browse for the location of the TrustStore on the computer, and in **Personal Certificate Store** field, browse for the location of the KeyStore on the computer.
The TrustStore and KeyStore contain the TLS certificates that are used with connections using client channel definition tables. It is possible that the TrustStore and KeyStore are in the same location on your computer.
6. (Optional) Click **Enter password...** in the Trusted certificate Store section to open the **SSL Password** dialog; in the **SSL Password** dialog, type the password that IBM MQ Explorer will need to access the store.
7. Click **Enter password...** in the Personal Certificate Store section to open the **SSL Password** dialog; in the **SSL Password** dialog, type the password that IBM MQ Explorer will need to access the store.
8. Click **OK** to save your changes and to close the Preferences dialog.

Results

IBM MQ Explorer can now use the TLS certificates in the TrustStore and KeyStore to connect to remote queue managers with an TLS-enabled connection.

Related tasks

[“Showing a remote queue manager” on page 99](#)

If you want to administer a remote queue manager, you must connect IBM MQ Explorer to the remote queue manager, so that the queue manager then appears in the Navigator view.

[“Creating a client channel definition table” on page 104](#)

You can create a client channel definition table for a queue manager to make it easier to connect instances of IBM MQ Explorer to the queue manager.

Related reference

[“Default security preferences” on page 192](#)

A security exit can be defined for all client connections in the same IBM MQ Explorer. This is known as a default security exit and the preferences for the security exit are described here.

Hiding queue managers

You can hide from view any queue manager that is displayed in the Navigator view. If you hide a queue manager that is a member of one or more queue manager Sets, the queue manager is not displayed in any of those Sets.

About this task

Hiding queue managers enables you to restrict the queue managers that are displayed in the Queue Managers folder if you have been working with many queue managers in IBM MQ Explorer.

To hide queue managers, complete the following steps.

Procedure

1. In the Navigator view, right-click the queue manager you want to hide.
 - Hold down the Ctrl key to select more than one queue manager then right-click
2. Click **Hide** to hide the queue manager.

Results

The selected queue managers are no longer displayed in the **Queue Managers** folder.

If the queue managers you have hidden are members of one or more queue manager Sets, then those queue managers are not displayed in those Sets.

Alternative procedure for hiding queue managers

Procedure

1. In the Navigator view, right-click the **Queue Managers** folder, then click **Show/Hide Queue Managers**. The Show/Hide Queue Managers dialog opens. A list of the visible queue managers is displayed in the **Shown Queue Managers** table of the Show/Hide Queue Managers dialog.
2. In the **Shown Queue Managers** table, click the name of the queue manager, then click **Hide**. The selected queue managers are now listed in the **Hidden Queue Managers** table.
3. Click **Close**.

Results

The queue manager is no longer displayed in the **Queue Managers** folder.

Hiding a queue manager using Sets

Before you begin

You can also hide from view any queue managers that are grouped in a queue manager Set. This enables you to restrict the queue managers that are displayed in the Set and Queue Managers folder if you have been working with many queue managers in IBM MQ Explorer.

Before you hide all the queue managers in a Set, the following steps must be taken:

1. You must display the queue manager sets as described in: [“Displaying queue manager sets” on page 233](#).
2. You must define a set for the queue managers as described in: [“Defining manual sets” on page 234](#) or: [“Defining automatic sets” on page 235](#).

About this task

To hide the queue managers in a Set, complete the following steps.

Procedure

- In the **Navigator** view, right-click the Set, then click **Hide All Queue Managers**.

Results

The queue manager is no longer displayed in the Set folder.

When you hide the queue managers in a Set, the queue managers are then hidden in every Set (including the **All** Set) not just the Set you selected.

Related tasks

[“Showing or hiding a queue manager” on page 98](#)

By default, the Navigator view shows all of the queue managers on the computer on which IBM MQ Explorer is installed. However, if you have any queue managers that you are not currently administering, you can, if you wish, choose to hide them. You can also show and hide remote queue managers.

[“Removing a queue manager” on page 108](#)

You can remove a queue manager from IBM MQ Explorer if you no longer want to administer it in IBM MQ Explorer.

Showing hidden queue managers

You can show queue managers that have previously been hidden from the Navigator view. You can restore all hidden queue managers together, or restore a specific queue manager. You can also show hidden queue managers that are grouped in a queue manager Set.

About this task

If you have hidden local or remote queue managers from view in the **Queue Managers** folder and you are now required to administer those queue managers, you can show the queue managers again.

To restore all hidden queue managers in one go, complete the following steps:

Procedure

In the **Navigator** view, right-click the **Queue Managers** folder, then click **Show All Hidden Queue Managers**

Results

All hidden queue managers are shown in the **Queue Managers** folder.

Showing specific hidden queue managers

About this task

To show specific hidden queue managers, complete the following steps.

Procedure

1. In the **Navigator** view, right-click the **Queue Managers** folder, then click **Show/Hide Queue Managers**. The Show/Hide Queue Managers dialog opens. A list of the hidden queue managers is displayed in the **Hidden Queue Managers** table of the **Show/Hide Queue Managers** dialog.
2. In the **Hidden Queue Managers** table, select one or more queue managers, then click **Show**. The selected queue managers are now listed in the **Shown Queue Managers** table.
3. Click **Close**.

Results

The selected queue managers are shown in the **Queue Managers** folder.

Showing a hidden queue manager using Sets

Before you begin

You can also show hidden queue managers that are grouped in a queue manager Set in IBM MQ Explorer.

Before you show the hidden queue managers in a Set, the following steps must be taken:

1. You must display the queue manager sets as described in: [“Displaying queue manager sets” on page 233](#).
2. You must define a set for the queue managers as described in: [“Defining manual sets” on page 234](#) or: [“Defining automatic sets” on page 235](#).

About this task

To show the hidden queue managers in a Set, complete the following steps.

Procedure

- In the **Navigator** view, right-click the Set, then click **Show All Queue Managers**.

Results

The previously hidden queue managers are now shown in the Set folder.

When you show the queue managers in a Set, the queue managers are then shown in every Set (including the **All** Set) not just the Set you selected.

Related tasks

[“Hiding queue managers” on page 106](#)

You can hide from view any queue manager that is displayed in the Navigator view. If you hide a queue manager that is a member of one or more queue manager Sets, the queue manager is not displayed in any of those Sets.

Removing a queue manager

You can remove a queue manager from IBM MQ Explorer if you no longer want to administer it in IBM MQ Explorer.

About this task

If you no longer want to administer a queue manager in IBM MQ Explorer, you can remove the queue manager from the **Queue Managers** folder.

To remove a queue manager:

Procedure

1. In the Navigator view, right-click the **Queue Managers** folder, then click **Show/Hide Queue Managers...**
The Show/Hide Queue Managers dialog opens.
2. If the queue manager is currently shown in the **Queue Managers** folder, in the **Shown Queue Managers** table, hide the queue manager so that the queue manager is displayed in the **Hidden Queue Managers** table.
For more information, see [Hiding queue managers](#).
3. In the **Hidden Queue Managers** table, click the name of the queue manager, then click **Remove...**
4. When prompted, click **Yes** to confirm that you want to remove the queue manager from IBM MQ Explorer.

Results

When you remove a queue manager from IBM MQ Explorer, the queue manager still exists on its host computer but you cannot administer it in IBM MQ Explorer until you add it to the **Queue Managers** folder again.

Related tasks

[“Showing or hiding a queue manager” on page 98](#)

By default, the Navigator view shows all of the queue managers on the computer on which IBM MQ Explorer is installed. However, if you have any queue managers that you are not currently administering, you can, if you wish, choose to hide them. You can also show and hide remote queue managers.

[“Administering remote queue managers” on page 111](#)

In IBM MQ Explorer, you can enable IBM MQ queue managers on a remote computer for remote administration.

Connecting or disconnecting a queue manager

If you want to administer a queue manager in IBM MQ Explorer, you must connect IBM MQ Explorer to the queue manager.

Before you begin

Before you can connect IBM MQ Explorer to a queue manager, perform the following tasks:

- Show the queue manager in the **Queue Managers** folder in IBM MQ Explorer.
- If the queue manager is on a different computer to IBM MQ Explorer, ensure that the queue manager is running.

About this task

To administer a queue manager in IBM MQ Explorer, you must connect IBM MQ Explorer to the queue manager. You can connect to any local queue manager, regardless of whether the queue manager is running. However, you can connect to a remote queue manager only if it is running.

You can also configure a queue manager so that IBM MQ Explorer automatically reconnects to it if the connection is lost. For more information, see [“Automatically reconnecting to a queue manager” on page 110](#)

Procedure

1. To connect IBM MQ Explorer to a queue manager: In the **Navigator** view, right-click the queue manager, then click **Connect** or **Disconnect**.

IBM MQ Explorer connects or disconnects the queue manager. The color of the queue manager's icon changes to yellow when connected, or gray when disconnected.

Disconnected queue managers remain in the **Queue Managers** folder. If you want to remove a queue manager completely from IBM MQ Explorer, see [“Removing a queue manager” on page 108](#).

2. If you have queue manager Sets enabled, then you can connect and disconnect all the queue managers in a Set: In the **Navigator** view, right-click the set, then click **Connect Queue Managers** or **Disconnect Queue Managers**.

All queue managers will be connected or disconnected depending on the option you selected.

Related tasks

[“Showing or hiding a queue manager” on page 98](#)

By default, the Navigator view shows all of the queue managers on the computer on which IBM MQ Explorer is installed. However, if you have any queue managers that you are not currently administering, you can, if you wish, choose to hide them. You can also show and hide remote queue managers.

Related reference

[“Icons in IBM MQ Explorer” on page 310](#)

IBM MQ Explorer uses icons to represent the different objects, such as queue managers, queues, and channels.

Automatically reconnecting to a queue manager

You can configure each queue manager so that IBM MQ Explorer automatically connects to it at startup or reconnects to it if the connection is lost; for example, if the network connection to a remote queue manager fails.

About this task

If you manually disconnect IBM MQ Explorer from the queue manager, the queue manager is not automatically reconnected until the next time that you close and restart IBM MQ Explorer. Only queue managers that are connected when IBM MQ Explorer is closed, and that are configured to autoreconnect, are automatically reconnected when IBM MQ Explorer is started again.

Procedure

- To configure a queue manager so that IBM MQ Explorer automatically reconnects to it, perform one of the following tasks:
 - For a remote queue manager, when you add the queue manager to IBM MQ Explorer, you can select the **Automatically connect to this queue manager at startup or if the connection is lost** check box in the Show/Hide Queue Managers wizard.
 - For local queue managers, and remote queue managers that are already shown in the **Queue Managers** folder, in the Navigator view, right-click the queue manager, then click **Autoreconnect**. A check mark is placed next to the menu item to indicate that IBM MQ Explorer is set to automatically reconnect to the queue manager if the connection is lost.

What to do next

To configure the queue manager so that IBM MQ Explorer does not automatically reconnect to it, right-click the queue manager, then click **Autoreconnect**. The check mark next to the menu item is removed.

Related tasks

[“Connecting or disconnecting a queue manager” on page 109](#)

If you want to administer a queue manager in IBM MQ Explorer, you must connect IBM MQ Explorer to the queue manager.

Administering remote queue managers

In IBM MQ Explorer, you can enable IBM MQ queue managers on a remote computer for remote administration.

About this task

In IBM MQ Explorer, you can administer IBM MQ on other computers that are connected to your computer by TCP/IP. It is possible to connect to a remote queue manager by using a different transport protocol. To use a different transport protocol, the connection must go through another queue manager to which IBM MQ Explorer is connected. Most operating system versions of IBM MQ can be administered remotely if they are at the command levels that are shown in the following table:

Operating system	Command level
UNIX, AIX®, HP-UX, Linux, and Solaris systems	500, 510, 520, 530, 600, 700, 701, 710, 750, 800, 900
IBM i	510, 520, 530, 600, 700, 701, 710, 800, 900
VMS	510
z/OS	600, 700, 701, 710, 800, 900
Windows systems	500, 510, 520, 530, 600, 700, 701, 710, 750, 800, 900
Tandem NSK	510
VSE/ESA™	510

For more information about operating systems and command levels, see [System Requirements for IBM MQ](#) on the external IBM website.

To find out what command level any IBM MQ queue manager supports, display the properties of the queue manager and check the CommandLevel (CMDLEVEL) property.

You cannot start, stop, create, or delete a remote queue manager from IBM MQ Explorer.

To administer a queue manager on Computer A from IBM MQ Explorer on Computer B:

Procedure

1. On Computer A, show the queue manager in IBM MQ Explorer.
2. On Computer A, start the queue manager.
3. To use the SYSTEM.ADMIN.SVRCONN server connection channel on Computer A to connect to the queue manager, enable the queue manager for remote administration.
4. On Computer B, show the remote queue manager in IBM MQ Explorer.

Results

You can administer the queue manager on Computer A from IBM MQ Explorer on Computer B.

Enabling remote administration of queue managers

In IBM MQ Explorer, you can administer queue managers that are hosted by other computers connected to your computer by TCP/IP. This includes queue managers that are hosted on z/OS.

About this task

It is possible to connect to a remote queue manager using a different transport protocol but the connection must go through another queue manager to which IBM MQ Explorer is connected.

To remotely administer a queue manager, the queue manager must be running and you must:

Procedure

1. Ensure that there is a running command server.
2. Create a server-connection channel to allow remote administration of the queue manager over TCP/IP.
3. Create a listener to accept incoming network connections.
4. Ensure that the listener is running.

Any TCP/IP listener and any server-connection channel can be used for this administration.

If you are planning to administer an IBM WebSphere MQ 6.0 or later queue manager from an IBM WebSphere MQ 5.3 computer, you must enable the IBM WebSphere MQ 6.0 or later queue manager for remote administration using the default SYSTEM.ADMIN.SVRCONN server connection channel. This is because IBM WebSphere MQ 5.3 must use the SYSTEM.ADMIN.SVRCONN server connection channel to connect to a remote queue manager.

If you are planning to administer an IBM WebSphere MQ 6.0 or later queue manager from an IBM WebSphere MQ 6.0 or later computer, you can either enable the IBM WebSphere MQ 6.0 or later queue manager for remote administration using the default SYSTEM.ADMIN.SVRCONN server connection channel, or specify a different server connection channel when the remote computer connects to the queue manager.

You can enable remote administration on a queue manager on Windows or Linux (x86 and x86-64 platforms) computers using IBM MQ Explorer. On other platforms, you must configure the queue manager from the command line.

For more information, see [Administering remote IBM MQ objects](#) or [Authority to administer IBM MQ on UNIX and Windows systems](#) in the IBM MQ online product documentation.

Enabling remote administration on an existing queue manager using the system default objects

In IBM MQ Explorer, you can administer queue managers that are hosted by other computers connected to your computer by TCP/IP using system default objects. This includes queue managers that are hosted on z/OS.

About this task

When you install IBM MQ, if there are queue managers on the computer from a previous installation, and any of the queue managers are not enabled for remote administration, you can choose to run the Remote Administration wizard. The Remote Administration wizard updates the queue managers that you specify.

If you have already installed IBM MQ on the remote Windows or Linux (x86 and x86-64 platforms) computer, and the computer hosts queue managers that are not enabled for remote administration, you can enable them for remote administration using the system default objects as follows:

Before you enable remote administration on an existing queue manager using the system default objects, start the queue manager in IBM MQ Explorer on the computer that hosts the remote queue manager.

To enable remote administration of an existing queue manager:

Procedure

1. Right-click the queue manager in the **Navigator** view, then click **Remote Administration....** The **Remote Administration** dialog opens. IBM MQ checks whether the SYSTEM.ADMIN.SVRCONN server connection channel exists, and checks whether there is a listener created and running. The results are displayed in the **Remote Administration** dialogue.
2. Click **Create** to create a SYSTEM.ADMIN.SVRCONN channel if one does not exist. The SYSTEM.ADMIN.SVRCONN channel is created.
3. Click **Create** to create a LISTENER.TCP listener if one does not exist. The LISTENER.TCP listener is created.
4. Click **Close** to close the dialog.

For more information, see [Authority to administer IBM MQ on UNIX and Windows systems](#) in the IBM MQ online product documentation.

Enabling remote administration when you create a new queue manager

When you create a new queue manager in IBM MQ Explorer, you can enable remote administration for this new queue manager. The queue manager is configured to use the SYSTEM.ADMIN.SVRCONN server connection channel for remote administration.

About this task

This task provides steps on how to enable remote administration when a new queue manager is created.

To enable a new queue manager for remote administration, complete the following steps.

Procedure

1. In the Create Queue Manager wizard, select the following options:
 - a) Create server connection channel
 - b) Create listener configured for TCP/IP
2. Type a port number in the **Listen on port number** field. The port number must not be in use by another running queue manager hosted on the same computer.

When the queue manager is created, it is configured to use the SYSTEM.ADMIN.SVRCONN server connection channel for remote administration.

For more information, see [Administering remote IBM MQ objects](#) and [Authority to administer IBM MQ on UNIX and Windows systems](#) in the IBM MQ online product documentation.

Maintaining intercommunications along message channels

You might sometimes need to take action in order to maintain intercommunications along message channels. For example you might need to resolve an in-doubt channel by either backing out or committing the messages, or reset channel synchronization if the message counts at the two ends of the channel are not in synchronization. You can also configure channels to reduce the possibility of a sending channel being put in doubt and made unavailable.

About this task

When a channel tries to commit a logical unit of work, if the receiving end of the channel is not available, the sending end of the channel is put in-doubt because it cannot determine whether the messages on the transmission queue have been committed. The messages are held on the transmission queue and no messages can be sent through the channel until the channel's status has been resolved. Often, in-doubt channels are automatically resolved by IBM MQ when connection between the two ends of the channel is re-established. However, this can cause delays, especially if the connection cannot be re-established because, for example, the receiving end of the channel has been deleted.

The Message Channel Agent (MCA) keeps a record of the number of messages sent and received (the Sequence Number) and the ID of the last-committed logical units of work (the LUWID).

- [“Resolving in-doubt channels” on page 115](#)
- [“Resetting channel synchronization” on page 114](#)
- [“Configuring the channel to reduce the opportunity of being put 'in-doubt'” on page 116](#)

For more information, see [Distributed queueing and clusters](#) in the IBM MQ online product documentation.

Related reference

[“Channel properties” on page 426](#)

You can set properties for all types of channels, including client-connection channels. Some properties are specific to certain types of channel.

Resetting channel synchronization

If synchronization errors are reported because the message counts at the two ends of the channel are not in synchronization, you can reset the synchronization.

About this task

The Message Channel Agents (MCAs) at the two ends of a channel each keep count of the number of messages sent through the channel so that synchronization between the two ends of the channel can be maintained. Synchronization can be lost, for example, when the channel definition at one end is deleted and then re-created. The re-created channel definition resets its count to 0, and if a queue manager attempts to use the channel, synchronization errors are reported because the two ends of the channel are not in synchronization.

To fix problems with channel synchronization, you must reset the count of the channel definition that was not re-created.

To reset the count, complete the following steps.

Procedure

1. In the Content view, right-click the channel definition that was not re-created, then click **Reset**. The Reset dialog opens.
2. In the Reset dialog, type the sequence number to which you want to reset the channel definition:
 - If the other end of the channel has been deleted and then re-created, type 0.
 - If the channel is a sender or server channel, type any number from 0 to the value defined in the Sequence number wrap attribute of the channel (the default value is 999,999,999). The new message sequence number is sent automatically to the other end of the channel, which then sets its number to match the next time that the channels are started.
 - For all other channel types, type the current sequence number of the other end of the channel. To find the current sequence number of the other end of the channel, right-click the name of the channel, then click **Status**.
3. Click **Yes** to reset the channel definition to the message count that you typed in the Message Sequence Number field.

Results

The two ends of the channel have the same message count and so are synchronized.

For more information, see [Distributed queueing and clusters](#) in the IBM MQ online product documentation.

Related tasks

[“Configuring queue managers and objects” on page 51](#)

You can configure many of the attributes of queue managers and their objects from IBM MQ Explorer using the properties dialogs.

[“Maintaining intercommunications along message channels” on page 113](#)

You might sometimes need to take action in order to maintain intercommunications along message channels. For example you might need to resolve an in-doubt channel by either backing out or committing the messages, or reset channel synchronization if the message counts at the two ends of the channel are not in synchronization. You can also configure channels to reduce the possibility of a sending channel being put in doubt and made unavailable.

Related reference

[“Channel properties” on page 426](#)

You can set properties for all types of channels, including client-connection channels. Some properties are specific to certain types of channel.

Resolving in-doubt channels

If there is no possibility that a link that has been lost might recover, you must resolve an in-doubt channel by either backing out or committing the messages.

About this task

The sending end of a channel might be holding messages in-doubt because, for example, it has lost the connection with the receiving end of the channel. If there is no prospect of the link recovering, the channel must be resolved either to back out the messages (restore the messages to the transmission queue) or commit the messages (discard the messages).

To resolve a channel, complete the following steps.

Procedure

1. Find out the last committed Logical Unit of Work ID (LUWID) for each end of the channel:
 - a) In the Content view, right-click the channel definition at one end of the channel, then click **Status...** The Status dialog for that channel definition opens.
 - b) In the Status dialog, look for the value in the **Last LUWID** column. This value shows the ID of the last logical unit of work that was committed by the channel. Make a note of the value.
 - c) Repeat Steps 1 and 2 for the channel definition at the other end of the channel.
2. In the Content view, right-click the sending end of the channel, then click **Resolve...** The Resolve dialog opens.
3. In the Resolve dialog, select the method with which to resolve the channel:
 - If the LUWID at the sending end of the channel is the same as the LUWID at the receiving end of the channel, click **Commit** to commit the messages and discard the messages from the transmission queue.
 - If the LUWID at the sending end of the channel is different from the LUWID at the receiving end of the channel, click **Back out** to back out the unit of work and retain the messages to the transmission queue so that the messages can be re-sent.

Results

The channel is no longer in doubt and the transmission queue can be used by a different channel to re-send the messages.

For more information, see [Distributed queueing and clusters](#) in the IBM MQ online product documentation.

Related tasks

[“Configuring queue managers and objects” on page 51](#)

You can configure many of the attributes of queue managers and their objects from IBM MQ Explorer using the properties dialogs.

[“Maintaining intercommunications along message channels” on page 113](#)

You might sometimes need to take action in order to maintain intercommunications along message channels. For example you might need to resolve an in-doubt channel by either backing out or committing the messages, or reset channel synchronization if the message counts at the two ends of the channel are not in synchronization. You can also configure channels to reduce the possibility of a sending channel being put in doubt and made unavailable.

Related reference

[“Channel properties” on page 426](#)

You can set properties for all types of channels, including client-connection channels. Some properties are specific to certain types of channel.

Configuring the channel to reduce the opportunity of being put 'in-doubt'

By using the **Batch heartbeat interval** attribute, you can reduce the possibility of a sending channel being put in doubt and made unavailable.

About this task

You can configure channels, using the **Batch heartbeat interval** attribute, so that the sending end of the channel checks that the receiving end of the channel is still active before the channel tries to commit the current logical unit of work. When the **Batch heartbeat interval** attribute is set, the sending end of the channel sends a heartbeat to the receiving end before the channel tries to commit the current logical unit of work.

If the sending channel has had a communication from the receiving channel within the **Batch heartbeat interval**, the receiving channel is assumed to be still active, otherwise a 'heartbeat' is sent to the receiving channel to check. The sending channel waits for a response from the receiving end of the channel for an interval, based on the number of seconds specified in the channel Heartbeat Interval (HBINT) attribute.

The advantage of using the **Batch heartbeat interval** is that instead of the sending channel being put in doubt and made unavailable, the only delay is the time during which the sending end of the channel is sending the heartbeat and waiting for a response from the receiving end of the channel.

To configure the **Batch heartbeat interval** attribute:

Procedure

1. Open the sending channel properties dialog.
2. On the **Extended** page, type the number of seconds that the sending end of the channel waits for a response from the receiving end of the channel.
3. Click **OK**.

Results

Whenever the channel is ready to commit a logical unit of work, the sending end of the channel sends a heartbeat to the receiving end of the channel to check that the receiving end of the channel is still active.

For more information, see [Distributed queueing and clusters](#) in the IBM MQ online product documentation.

Related tasks

[“Configuring queue managers and objects” on page 51](#)

You can configure many of the attributes of queue managers and their objects from IBM MQ Explorer using the properties dialogs.

[“Maintaining intercommunications along message channels” on page 113](#)

You might sometimes need to take action in order to maintain intercommunications along message channels. For example you might need to resolve an in-doubt channel by either backing out or committing the messages, or reset channel synchronization if the message counts at the two ends of the channel are not in synchronization. You can also configure channels to reduce the possibility of a sending channel being put in doubt and made unavailable.

Related reference

[“Channel properties” on page 426](#)

You can set properties for all types of channels, including client-connection channels. Some properties are specific to certain types of channel.

Configuring publish/subscribe messaging

In publish/subscribe messaging, the sender of a message (the publisher) is decoupled from the recipient of the message (the subscriber) so that the publisher does not need to know who will receive the message and the subscriber does not necessarily know who sent the message. The publisher publishes the message to the broker, which is then responsible for distributing the message to any subscribers who have registered an interest in the information in the message.

Procedure

- [“Publishers and subscribers” on page 117](#)
- [Configuring publish/subscribe messaging for IBM WebSphere MQ 7.0 and later queue managers.](#)

Publishers and subscribers

Publishers and subscribers are applications that send and receive messages (publications) using the publish/subscribe method of messaging. Publishers and subscribers are decoupled from one another so that publishers do not know the destination of the information that they send, and subscribers do not know the source of the information that they receive.

The provider of the information is called a *publisher*. Publishers supply information about a subject without needing to know anything about the applications that are interested in the information.

The consumer of the information is called a *subscriber*. The subscriber decides what information it is interested in, and then waits to receive that information. Subscribers can receive information from many different publishers, and the information they receive can also be sent to other subscribers.

The information is sent in an IBM MQ message, and the subject of the information is identified by a *topic string*. The publisher specifies the topic string when it publishes the information, and the subscriber specifies the topic strings on which it wants to receive publications. The subscriber is sent information about only those topic strings to which it subscribes.

IBM WebSphere MQ 7.0 and later queue managers, use a Pub/Sub Engine instead of a broker to control the interactions between publishers and subscribers. The Pub/Sub Engine receives messages from publishers, and subscription requests from subscribers. The Pub/Sub Engine's job is to route the published data to the target subscribers.

Related concepts

[“Topics” on page 30](#)

A topic identifies what a publication is about. A topic is a character string that describes the subject of the information that is published in a Publish/Subscribe message. As a subscriber, you can specify a topic or range of topics by using wildcards to receive the information that you require.

[“Publications” on page 33](#)

Publications are messages that are sent by an application to the Publish/Subscribe Engine. The Publish/Subscribe Engine then sends the messages on to any applications that have subscribed to receive the messages.

Related tasks

[“Configuring publish/subscribe for IBM WebSphere MQ 7.0 and later queue managers” on page 129](#)

In IBM MQ Explorer, you can configure IBM WebSphere MQ 7.0 and later queue managers as Publish/Subscribe Engines to route messages between publishing applications and subscribing applications. To test your configurations, you can register as a subscriber, and send and receive test publications if you are authorized to do so.

Configuring publish/subscribe for IBM WebSphere MQ 6.0 queue managers

In IBM MQ Explorer, you can configure IBM WebSphere MQ 6.0 queue managers as brokers to route messages between publishing applications and subscribing applications. To test your configurations, you can register as a publisher and as a subscriber, and send and receive test publications.

Before you begin

Before you start:

- [“Creating and configuring queue managers and objects” on page 27](#). The queue manager will host the publish/subscribe broker.

For more information about brokers and broker networks, see [Publish/subscribe messaging](#) in the IBM WebSphere MQ online product documentation.

About this task

To configure publish/subscribe messaging on an IBM WebSphere MQ 6 queue manager:

Procedure

1. [Register as a publisher](#).
2. [Register as a subscriber](#).
3. [Send and receive test publications](#).

What to do next

In IBM MQ Explorer, you can also view and clear retained publications, and view a list of the publishing applications, the subscribing applications, and the streams.

Related tasks

[“Viewing retained publications” on page 126](#)

Retained publications are publications that are retained by the broker after sending the publication to interested subscribers (usually, the broker deletes publications after sending them and does not retain copies). You can view the retained publication that is currently kept by the broker. The broker retains only one publication for each topic.

[“Viewing a list of registered publishers” on page 124](#)

You can view a list of applications that are registered to publish on topics on a broker, or a list of applications that are registered to publish on a specific topic.

[“Viewing a list of subscribers” on page 124](#)

You can view a list of applications that are subscribed to topics on a broker, or a list of applications that are subscribed to a specific topic.

[“Viewing a list of streams” on page 125](#)

Streams provide a way of separating the flow of information for different topics. You can view a list of all the streams that are currently available on the broker.

Topics

A topic identifies what a publication is about. Topic names are characters strings. Subscriptions can include wildcard characters to retrieve the information required by the application. IBM MQ Publish/Subscribe recognizes the asterisk (*) and the question mark (?) as wildcard characters.

Topic names

Topic names are strings of characters. You could have high-level topics named 'Sport', 'Stock', 'Films', and 'TV', and you could divide the 'Sport' topic into separate, more specific topics covering different sports; for example:

```
Sport/Soccer Sport/Golf Sport/Tennis
```

These topics could then be divided further, to separate different types of information about each sport:

```
Sport/Soccer/Fixtures Sport/Soccer/Results Sport/Soccer/Reports
```

IBM MQ Publish/Subscribe does not recognize that the forward slash (/) character is being used in a special way but if you use the forward slash (/) character as a separator, you can ensure compatibility with other WebSphere business integration applications.

You can use any character in the single-byte character set for which the machine is configured in a character string. Consider, however, whether the topic string might need to be translated to a different character representation, in which case you must use only those characters that are available in the configured character set of all relevant machines.

Topic strings are case sensitive, and a blank character has no special meaning. As a subscriber, you can specify a topic or range of topics using wildcards to receive the information in which you are interested.

Wildcards in topic strings

IBM MQ Publish/Subscribe recognizes the following wildcard characters:

Wildcard character	Meaning
Asterisk (*)	Zero or more characters
Question mark (?)	One character

For example, you could use the following topic strings in subscriptions to retrieve particular sets of information:

All information on Sport, Stock, Films, and TV.

Sport/*

All information on Soccer, Golf, and Tennis.

Sport/Soccer/*

All information on Soccer (Fixtures, Results, and Reports).

Sport/*/Results

All Results for Soccer, Golf, and Tennis.

If you want to include the asterisk (*) or question mark (?) in the topic string of a subscription, you must use the percent sign (%) as an escape character. For example, if you want to subscribe to a topic that is named ABC*D, you must enter the string ABC%D.

If you want to use a % character in the topic string of a subscription, you must specify two percent signs (%%). A percent sign (%) in a topic string must always be followed by an asterisk (*), a question mark (?), or another percent sign (%).

Wildcards do not span streams.

Related concepts

[“Publishers and subscribers” on page 117](#)

Publishers and subscribers are applications that send and receive messages (publications) using the publish/subscribe method of messaging. Publishers and subscribers are decoupled from one another so that publishers do not know the destination of the information that they send, and subscribers do not know the source of the information that they receive.

[“Streams” on page 121](#)

Streams apply to IBM WebSphere MQ 6.0 queue managers only, and provide a way of separating the flow of information for different topics. A stream is implemented as a set of queues, one at each broker that supports the stream. Each queue has the same name (the name of the stream). The default stream set up between all the brokers in a network is called `SYSTEM.BROKER.DEFAULT.STREAM`.

Related reference

[“Topic status attributes” on page 624](#)

The status attributes of topics.

Publications

Publications are messages that are sent by an application to the broker. The broker then sends the messages on to any applications that have subscribed to receive the messages.

Note: This information is for IBM WebSphere MQ 6.0 queue managers only.

The broker can handle publications that it receives in different ways, depending on the type of information contained in the publication.

State and event information

Publications can be categorized by the type of information that they contain:

State publications

State publications contain information about the current *state* of something, such as the price of stock or the current score at a soccer match. When something happens (for example, the stock price changes or the soccer score changes), the previous state information is no longer required because it is superseded by the new information.

A subscriber application wants to receive the current version of the state information on startup, and to be sent new information whenever the state changes.

Event publications

Event publications contain information about individual *events* that occur, such as a trade in some stock or the scoring of a particular goal. Each event is independent of other events.

A subscriber wants to receive information about events as they happen.

Retained publications

By default, when a broker has sent a publication to all interested subscribers, the broker deletes the publication. This type of processing is suitable for event information but is not always suitable for state information. A publisher can specify that the broker must keep a copy of a publication, which is then called a *retained publication*. The copy can be sent to subsequent subscribers who register an interest in the topic. This means that new subscribers do not have to wait for information to be published again before they receive it. For example, a subscriber who registers a subscription to a stock price would receive the current stock price straightaway, without waiting for the stock price to change (and, therefore, be re-published).

The broker retains only one publication for each topic, so the old publication is deleted when a new one arrives. So, ensure that only one publisher is sending retained publications on each topic.

Subscribers can specify that they do not want to receive retained publications, and existing subscribers can ask for duplicate copies of retained publications to be sent to them.

Related concepts

[“Publishers and subscribers” on page 117](#)

Publishers and subscribers are applications that send and receive messages (publications) using the publish/subscribe method of messaging. Publishers and subscribers are decoupled from one another so that publishers do not know the destination of the information that they send, and subscribers do not know the source of the information that they receive.

[“Topics” on page 119](#)

A topic identifies what a publication is about. Topic names are characters strings. Subscriptions can include wildcard characters to retrieve the information required by the application. IBM MQ Publish/Subscribe recognizes the asterisk (*) and the question mark (?) as wildcard characters.

Related tasks

[“Registering as a publisher” on page 121](#)

The first time that you publish on a topic, you are implicitly registered with the broker as a publisher. If, however, the broker is not aware of the stream on which you publish, or if you do not know whether the broker is aware of the stream, you must explicitly register as a publisher so that the broker is aware of the stream.

Streams

Streams apply to IBM WebSphere MQ 6.0 queue managers only, and provide a way of separating the flow of information for different topics. A stream is implemented as a set of queues, one at each broker that supports the stream. Each queue has the same name (the name of the stream). The default stream set up between all the brokers in a network is called `SYSTEM.BROKER.DEFAULT.STREAM`.

Streams can be created by an application or by the administrator. Stream names are case-sensitive, and stream queues must be local queues (not alias queues). Stream names beginning with the characters `'SYSTEM.BROKER.'` are reserved for IBM WebSphere MQ use.

A broker has a separate thread for each stream that it supports. If multiple streams are used, the broker can process publications arriving at different stream queues in parallel. Streams can also provide high-level grouping of topics, can restrict the range of publications and subscriptions that a broker has to deal with, can provide access control, can define a certain quality of service for broker-to-broker communication of publications, and can allow different queue attributes to be assigned for publication on different streams.

Related concepts

[“Topics” on page 119](#)

A topic identifies what a publication is about. Topic names are characters strings. Subscriptions can include wildcard characters to retrieve the information required by the application. IBM MQ Publish/Subscribe recognizes the asterisk (*) and the question mark (?) as wildcard characters.

[“IBM MQ queues” on page 29](#)

A queue is a container for messages. Business applications that are connected to the queue manager that hosts the queue can retrieve messages from the queue or can put messages on the queue.

Registering as a publisher

The first time that you publish on a topic, you are implicitly registered with the broker as a publisher. If, however, the broker is not aware of the stream on which you publish, or if you do not know whether the broker is aware of the stream, you must explicitly register as a publisher so that the broker is aware of the stream.

Before you begin

Note: This information is for IBM WebSphere MQ 6.0 queue managers only.

Before you start:

- [Show the queue manager that hosts the broker.](#)

About this task

To register with the broker as a publisher:

Procedure

1. In the Navigator view, expand the queue manager that hosts the broker with which you want to register as a publisher, then click the **Topics** folder.

The existing topics on the broker are shown in the Content view.

2. Start the Register Publisher wizard:

- If the topic is listed in the Content view, right-click the topic then click **Register Publisher...**
- If the topic is not listed in the Content view, right-click the **Topics** folder, then click **Register Publisher...**

The Register Publisher wizard opens.

3. Work through the wizard to register as a publisher on the topic on which you want to publish messages.

Results

The Content view updates to show the total number of publishers that are registered to publish on the topic.

What to do next

Next:

1. [View a list of registered publishers.](#)
2. [Send and receive test publications.](#)

Related concepts

[“Topics” on page 119](#)

A topic identifies what a publication is about. Topic names are characters strings. Subscriptions can include wildcard characters to retrieve the information required by the application. IBM MQ Publish/Subscribe recognizes the asterisk (*) and the question mark (?) as wildcard characters.

[“Streams” on page 121](#)

Streams apply to IBM WebSphere MQ 6.0 queue managers only, and provide a way of separating the flow of information for different topics. A stream is implemented as a set of queues, one at each broker that supports the stream. Each queue has the same name (the name of the stream). The default stream set up between all the brokers in a network is called SYSTEM.BROKER.DEFAULT.STREAM.

[“Publishers and subscribers” on page 117](#)

Publishers and subscribers are applications that send and receive messages (publications) using the publish/subscribe method of messaging. Publishers and subscribers are decoupled from one another so that publishers do not know the destination of the information that they send, and subscribers do not know the source of the information that they receive.

Registering as a subscriber

To register your interest in receiving publications, you must register with the broker as a subscriber on the topics in which you are interested.

Before you begin

Note: This information is for IBM WebSphere MQ 6.0 queue managers only.

Before you start:

- [Show the queue manager that hosts the broker in IBM MQ Explorer.](#)

About this task

To register as a subscriber:

Procedure

1. Ensure that you have the following object authorities:

Object	Authority
The broker's control queue (SYSTEM.BROKER.CONTROL.QUEUE)	Put
The stream queues to which you publish messages	Browse
The subscriber queue that will receive the publications	Put

2. In the Navigator view, expand the queue manager that hosts the broker with which you want to register as a subscriber, then click the **Topics** folder.
The existing topics on the broker are shown in the Content view.
3. Start the Register Subscriber wizard:
 - If the topic already exists, right-click the topic then click **Register Subscriber...**
 - If the topic does not already exist, right-click the **Topics** folder, then click **Register Subscriber...**The Register Subscriber wizard opens.
4. Work through the wizard to register as a subscriber on the topic on which you want to receive messages.

Results

The Content view updates to show the total number of publishers that are registered to publish on the topic.

What to do next

Next:

1. [View a list of registered subscribers.](#)
2. [Send and receive test publications.](#)

Related concepts

[“Authorities you can set on IBM MQ objects” on page 182](#)

You can set authorities for users and groups accessing different IBM MQ objects.

[“Topics” on page 119](#)

A topic identifies what a publication is about. Topic names are characters strings. Subscriptions can include wildcard characters to retrieve the information required by the application. IBM MQ Publish/Subscribe recognizes the asterisk (*) and the question mark (?) as wildcard characters.

[“Publishers and subscribers” on page 117](#)

Publishers and subscribers are applications that send and receive messages (publications) using the publish/subscribe method of messaging. Publishers and subscribers are decoupled from one another so that publishers do not know the destination of the information that they send, and subscribers do not know the source of the information that they receive.

Related tasks

[“Granting authorities on a specific object” on page 169](#)

A user must have the correct authorities to perform operations on objects; for example, to browse the messages on a queue.

Viewing a list of registered publishers

You can view a list of applications that are registered to publish on topics on a broker, or a list of applications that are registered to publish on a specific topic.

Before you begin

Note: This information is for IBM WebSphere MQ 6.0 queue managers only.

Before you start:

- [Register as a publisher on a topic.](#)

About this task

To view a list of registered publishers:

Procedure

1. In the Navigator view, expand the queue manager that hosts the broker that you want to register with as a publisher, then click the **Topics** folder.
The existing topics on the broker are shown in the Content view.
2. Request a list of registered publishers:
 - If you want to view all the applications that are registered to publish on a specific topic, right-click the topic, then click **Registered Publishers...**
 - If you want to view all the applications that are registered to publish on a topic on the broker, right-click the **Topics** folder, then click **View Registered Publishers...**

Results

The Registered Publishers dialog opens, listing details of the applications, including the user name under which the application is running, that are registered as publishers.

Related concepts

[“Topics” on page 119](#)

A topic identifies what a publication is about. Topic names are characters strings. Subscriptions can include wildcard characters to retrieve the information required by the application. IBM MQ Publish/Subscribe recognizes the asterisk (*) and the question mark (?) as wildcard characters.

Related reference

[“IBM MQ Explorer Content view” on page 324](#)

The Content view in IBM MQ Explorer displays information about objects and properties.

[“IBM MQ Explorer Navigator view” on page 316](#)

The Navigator view in IBM MQ Explorer displays all of the IBM MQ objects that you can administer and monitor in IBM MQ Explorer.

Viewing a list of subscribers

You can view a list of applications that are subscribed to topics on a broker, or a list of applications that are subscribed to a specific topic.

Before you begin

Note: This information is for IBM WebSphere MQ 6.0 queue managers only.

Before you start:

- [Register as a subscriber on a topic.](#)

About this task

To view a list of subscribers:

Procedure

1. In the **Navigator** view, expand the queue manager that hosts the broker with which you want to register as a subscriber, then click the **Topics** folder.
The existing topics on the broker are shown in the **Content** view.
2. Request a list of subscribers:
 - If you want to view all the applications that are subscribed to a specific topic, right-click the topic, then click **Subscribers...**
 - If you want to view all the applications that are subscribed to a topic on the broker, right-click the **Topics** folder, then click **View Subscribers...**

Results

The **Registered Subscribers** dialog opens, listing details of the applications, including the user name under which the application is running, that are subscribed.

Related concepts

[“Topics” on page 119](#)

A topic identifies what a publication is about. Topic names are characters strings. Subscriptions can include wildcard characters to retrieve the information required by the application. IBM MQ Publish/Subscribe recognizes the asterisk (*) and the question mark (?) as wildcard characters.

Related reference

[“IBM MQ Explorer Content view” on page 324](#)

The Content view in IBM MQ Explorer displays information about objects and properties.

[“IBM MQ Explorer Navigator view” on page 316](#)

The Navigator view in IBM MQ Explorer displays all of the IBM MQ objects that you can administer and monitor in IBM MQ Explorer.

Viewing a list of streams

Streams provide a way of separating the flow of information for different topics. You can view a list of all the streams that are currently available on the broker.

Before you begin

Note: This information is for IBM WebSphere MQ 6.0 queue managers only.

If you do not specify a stream, the default stream is used. The default stream set up between all the brokers in a network is called SYSTEM.BROKER.DEFAULT.STREAM

Before you start:

- [Show the queue manager that hosts the broker.](#)

About this task

To view a list of streams on the broker:

Procedure

1. In the Navigator view, expand the queue manager that hosts the broker.
2. Right-click the **Topics** folder, then click **View Streams...**

Results

The Streams dialog opens, listing all the streams on the broker.

Related concepts

[“Streams” on page 121](#)

Streams apply to IBM WebSphere MQ 6.0 queue managers only, and provide a way of separating the flow of information for different topics. A stream is implemented as a set of queues, one at each broker that supports the stream. Each queue has the same name (the name of the stream). The default stream set up between all the brokers in a network is called SYSTEM.BROKER.DEFAULT.STREAM.

Viewing retained publications

Retained publications are publications that are retained by the broker after sending the publication to interested subscribers (usually, the broker deletes publications after sending them and does not retain copies). You can view the retained publication that is currently kept by the broker. The broker retains only one publication for each topic.

About this task

Note: This information is for IBM WebSphere MQ 6.0 queue managers only.

To view a retained publication:

Procedure

1. In the Navigator view, expand the queue manager that hosts the broker, then click the **Topics** folder.
The existing topics on the broker are shown in the Content view.
2. In the Content view, right-click the topic on which the retained publication was published, then click **View Retained Publication...**

Results

The Retained Messages properties dialog opens displaying information about the retained publication.

Related concepts

[“Publications” on page 120](#)

Publications are messages that are sent by an application to the broker. The broker then sends the messages on to any applications that have subscribed to receive the messages.

[“Topics” on page 119](#)

A topic identifies what a publication is about. Topic names are characters strings. Subscriptions can include wildcard characters to retrieve the information required by the application. IBM MQ Publish/Subscribe recognizes the asterisk (*) and the question mark (?) as wildcard characters.

Related tasks

[“Clearing retained publications” on page 126](#)

Retained publications are publications that are retained by the broker after sending the publication to interested subscribers (usually, the broker deletes publications after sending them and does not retain copies). You can clear the retained publication that is currently kept by the broker. The broker retains only one publication for each topic.

Clearing retained publications

Retained publications are publications that are retained by the broker after sending the publication to interested subscribers (usually, the broker deletes publications after sending them and does not retain copies). You can clear the retained publication that is currently kept by the broker. The broker retains only one publication for each topic.

Before you begin

Note: This information is for IBM WebSphere MQ 6.0 queue managers only.

Before you start:

- [Register as a publisher on a topic on the broker.](#)

- [Publish a test retained publication on the topic.](#)

About this task

To clear a retained publication:

Procedure

1. In the **Navigator** view, expand the queue manager that hosts the broker, then click the **Topics** folder.
The existing topics on the broker are shown in the **Content** view.
2. In the **Content** view, right-click the topic on which the retained publication was published, then click **Clear Retained Publication...**

Results

The **Clear Retained Publication** confirmation dialog opens asking you to confirm that you want to clear the retained publication. Click **Yes** to clear the retained publication.

What to do next

Related concepts

[“Publications” on page 120](#)

Publications are messages that are sent by an application to the broker. The broker then sends the messages on to any applications that have subscribed to receive the messages.

[“Topics” on page 119](#)

A topic identifies what a publication is about. Topic names are characters strings. Subscriptions can include wildcard characters to retrieve the information required by the application. IBM MQ Publish/Subscribe recognizes the asterisk (*) and the question mark (?) as wildcard characters.

Related tasks

[“Viewing retained publications” on page 126](#)

Retained publications are publications that are retained by the broker after sending the publication to interested subscribers (usually, the broker deletes publications after sending them and does not retain copies). You can view the retained publication that is currently kept by the broker. The broker retains only one publication for each topic.

Sending and receiving test publications

You can send (publish) and receive (subscribe to) test publications (messages) to check that your broker network and topics work as intended. You can configure a publication so that the broker retains a copy after publishing it to the subscribers. This enables new subscribers to receive the publication even if they subscribed after it was published.

Before you begin

Note: This information is for IBM WebSphere MQ 6.0 queue managers only.

Before you start sending and receiving test publications:

- [Show the queue manager that hosts the broker.](#)

About this task

To send and receive test publications:

Procedure

1. Subscribe to the topic that you want to test:
 - a) In the Navigator view, expand the queue manager that hosts the broker.
 - b) Right-click the **Topics** folder, then click **Test Subscription...**

- The Subscribe application opens.
2. Publish a message to the same topic:
 - a) In the Navigator view, expand the queue manager that hosts the broker.
 - b) Right-click the **Topics** folder, then click **Test Publication...**
The Publish Test Message application opens.
 - c) Optional: In the **Stream** field, select a different stream.
 - d) In the **Topic** field, type the name of the topic on which you want to publish the message.
You or another publisher can already be registered to publish on the topic, or you can enter a new topic name. When you publish the message, you are automatically registered as a publisher on the topic.
 - e) In the **Message data** field, type a message to send in the publication.
For example, type Hello, world!
 - f) Click **Publish message** to send the message to the broker.The subscriber receives the message (the publication).
 3. Start another instance of the Subscribe application.
The second Subscribe application does not receive the message that was published by the Publish Test Message application because it was not subscribing to the topic at the time that the publication was sent to the broker.
 4. Unsubscribe the second Subscribe instance from the topic.
 - a) In the second Subscribe application, click **Unsubscribe**.
The second Subscribe application can no longer receive publications on that topic. The first Subscribe application can still receive publications on that topic.
 5. Publish a retained publication to the topic.
 - a) In the Publish application, select the **Retained message** check box.
 - b) Change the text in the **Message data** field.
For example, type Hi, I'm home.
 - c) Click **Publish message**.
The retained publication is published to the broker. The first Subscribe application receives the retained publication. The second Subscribe application does not receive the publication because it is currently not subscribed.
 6. Subscribe the second Subscribe application to the topic again:
 - a) In the second Subscribe application, click **Subscribe**.
The second Subscribe application is subscribed to the topic again and receives the retained publication because the broker kept a copy of the retained publication.

Results

You have now published and subscribed to test publications, including retained publications.

Related concepts

[“Publications” on page 120](#)

Publications are messages that are sent by an application to the broker. The broker then sends the messages on to any applications that have subscribed to receive the messages.

[“Topics” on page 119](#)

A topic identifies what a publication is about. Topic names are characters strings. Subscriptions can include wildcard characters to retrieve the information required by the application. IBM MQ Publish/Subscribe recognizes the asterisk (*) and the question mark (?) as wildcard characters.

Configuring publish/subscribe for IBM WebSphere MQ 7.0 and later queue managers

In IBM MQ Explorer, you can configure IBM WebSphere MQ 7.0 and later queue managers as Publish/Subscribe Engines to route messages between publishing applications and subscribing applications. To test your configurations, you can register as a subscriber, and send and receive test publications if you are authorized to do so.

Before you begin

If you require more conceptual information about Publish/Subscribe, Topics, Subscriptions, and Publications, then refer to one of the following links in the online product documentation:

- [Publish/subscribe messaging](#)
- [Topics](#)
- [Subscribers and subscriptions](#)
- [Publishers and publications](#)

Before you begin configuring:

- [“Creating and configuring queue managers and objects” on page 27](#). The queue manager will host the Publish/Subscribe Engine.

About this task

To configure publish/subscribe messaging on an IBM WebSphere MQ 7.0 or later queue manager, complete one or more of the tasks described in the subtopics.

Procedure

- [“Creating a new topic” on page 130](#)
- [“Creating a new cluster topic” on page 130](#)
- [“Viewing the topic status” on page 132](#)
- [“Sending and receiving test publications on a topic object folder” on page 132](#)
- [“Sending and receiving test publications for specific topics” on page 134](#)
- [“Viewing topic status for publishers” on page 135](#)
- [“Viewing topic status for subscribers” on page 136](#)
- [“Creating a new subscription” on page 137](#)
- [“Viewing a list of subscribers” on page 138](#)
- [“Refreshing proxy subscriptions” on page 138](#)
- [“Creating a new Multicast communication information object” on page 139](#)

What to do next

In IBM MQ Explorer, you can also view and clear retained publications.

Creating a new topic

A topic identifies what a publication is about. A topic is a character string that describes the subject of the information that is published in a publish/subscribe message.

Before you begin

For the latest information on topic strings, wildcard characters, special characters, and topic trees, see the following links.

- A topic string can include any character from the Unicode character set, including the space character. However, there are characters that have special meanings. The characters; plus sign (+), number sign (#), asterisk (*), and question mark (?) are described in [Wildcard schemes](#) in the IBM MQ online product documentation.
- Topic strings are case sensitive, and although a null character does not cause an error, do not use null characters in your topic strings. For the latest information on topic strings, see [Using topic strings](#) in the IBM MQ online product documentation.
- Each topic that you define is an element, or node, in the topic tree. For the latest information on topic trees, see [Topic trees](#) in the IBM MQ online product documentation.

The queue manager that hosts the Publish/Subscribe engine must be visible in the **Navigator** view. To show the queue manager, follow the instructions in [“Showing or hiding a queue manager” on page 98](#)

About this task

To create a new topic in the IBM MQ Explorer:

Procedure

1. Expand the queue manager that hosts the Publish/Subscribe Engine to display the object-folders in the **Navigator** view.
2. Right-click **Topics**, then click **New > Topic**.

Results

The **New Topic** wizard opens. Work through the wizard to create a new topic.

What to do next

For information about topic names, topic strings and topic wildcards, refer to the following links.

Related concepts

[“Topics” on page 30](#)

A topic identifies what a publication is about. A topic is a character string that describes the subject of the information that is published in a Publish/Subscribe message. As a subscriber, you can specify a topic or range of topics by using wildcards to receive the information that you require.

Related tasks

[“Viewing the topic status” on page 132](#)

A topic identifies what a publication is about. The topic status displays information about the topic, such as publications and subscriptions.

Creating a new cluster topic

You can either convert an existing topic into a cluster topic, or create a new cluster topic. You do this by navigating to the topic properties, then specifying the name of the cluster that is to host this topic, and the cluster routing mechanism to use for publications on this topic.

Before you begin

Create a cluster containing two or more queue managers, as described in [“Creating a queue manager cluster” on page 145](#).

The `SYSTEM.BASE.TOPIC` and `SYSTEM.DEFAULT.TOPIC` should not typically be used as cluster topics. This is because `SYSTEM.BASE.TOPIC` is present on all cluster queue managers, therefore it only affects the local queue manager unless it is modified on all queue managers to have a correctly functioning publish/subscribe cluster. The entire topic tree is also within a single cluster, preventing subsections of topic space being in a cluster, and making it harder to divide subsections of topic space into separate clusters. However, there are situations where it is necessary, such as IBM Integration BusCollective migration. For more information, see [Publish/subscribe messaging](#) in the online product documentation.

There are also a number of reasons not to make `SYSTEM.DEFAULT.TOPIC` a cluster topic: it is present on all queue managers in a cluster, so it only affects the local queue manager, and all topics defined while it is a cluster topic also become cluster topics in the same cluster.

About this task

To create a new cluster topic in the IBM MQ Explorer **Navigator** view, complete the following steps:

Procedure

1. Expand the cluster queue manager in which you want to create a new cluster topic.
2. In the navigation pane, select **Topics**.
A list of existing topics is displayed in the main pane.
3. Either select an existing topic, or create a new topic.
 - To select an existing topic, double-click the topic in the main pane.
 - To create a new topic, right-click **Topics** in the navigation pane, then select **New > Topic**. For more information, see [“Creating a new topic” on page 130](#).
4. In the properties pane, click **Cluster** to open the **Cluster** properties page.
5. Type the name of the cluster you want the topic to belong to in the **Cluster topic** field.
6. Optional: For IBM MQ 8.0 and later versions, select the routing mechanism from the **Cluster route** drop-down list.

The choices are as follows:

Direct

Messages published on one queue manager are sent directly from that queue manager to every subscription on any other queue manager in the cluster.

Topic host

Messages published on one queue manager are sent from there to a queue manager that hosts the definition of the topic. That *topic host queue manager* routes the message on to every subscription on any other queue manager in the cluster.

7. Click **Apply** to save the change.

Results

The topic has now become a cluster topic.

Related concepts

[“Cluster topics” on page 31](#)

Topics can be clustered in a similar manner to cluster queues, although an individual topic object can be a member of only one cluster. A topic is made into a cluster topic by defining, on the topic object, the name of the cluster that is to host the topic, and the cluster routing mechanism to use for publications on this topic.

Related tasks

[“Viewing the topic status” on page 132](#)

A topic identifies what a publication is about. The topic status displays information about the topic, such as publications and subscriptions.

Viewing the topic status

A topic identifies what a publication is about. The topic status displays information about the topic, such as publications and subscriptions.

Before you begin

Before you start:

- [Show the queue manager that hosts the Publish/Subscribe Engine.](#)

About this task

To view the status of a topic in the IBM MQ Explorer:

Procedure

1. In the **Navigator** view, expand the queue manager that hosts the Publish/Subscribe Engine, then click the **Topics** folder. The existing topics on the Publish/Subscribe Engine are shown in the **Content** view.
2. In the **Content** view, right-click the topic that you want to view the status for, then click **Status**.

Results

The **Status** dialog opens. One pane of the **Status** dialog shows the Topic String tree structure. You can expand and collapse the topic string to navigate the tree structure and display individual topic status.

What to do next

For information about topic names, topic strings, and topic properties, refer to the topics linked at the end of this topic.

Related concepts

[“Topics” on page 30](#)

A topic identifies what a publication is about. A topic is a character string that describes the subject of the information that is published in a Publish/Subscribe message. As a subscriber, you can specify a topic or range of topics by using wildcards to receive the information that you require.

Related tasks

[“Creating a new topic” on page 130](#)

A topic identifies what a publication is about. A topic is a character string that describes the subject of the information that is published in a publish/subscribe message.

Related reference

[“Topic properties” on page 458](#)

An IBM MQ topic is an IBM MQ object that identifies what a publication is about. You can set attributes for topics. Some topic attributes are specific to z/OS topics. Also, there are some attributes that you can alter only while you are creating a topic. You cannot modify these attributes after the IBM MQ topic has been created.

[“Topic status attributes” on page 624](#)

The status attributes of topics.

Sending and receiving test publications on a topic object folder

You can send (publish) and receive (subscribe to) test publications (messages) to check that your Publish/Subscribe Engine network and topics work as intended. You can configure a publication so that the Publish/Subscribe Engine retains a copy after publishing it to the subscribers. This enables new subscribers to receive the publication even if they subscribed after it was published.

Before you begin

Before you start:

- [Show the queue manager that hosts the Pub/Sub Engine.](#)

About this task

To send and receive test publications for any topic:

Procedure

1. Subscribe to the topic that you want to test:
 - a) In the Navigator view, expand the queue manager that hosts the Publish/Subscribe Engine.
 - b) Right-click the **Topics** folder, then click **Test Subscription....**
The **Subscribe** application opens.
 - c) Type a topic string in the Topic String field. The topic string must be the same name as the publisher.
2. Publish a message to the same topic:
 - a) In the Navigator view, expand the queue manager that hosts the Publish/Subscribe Engine.
 - b) Right-click the **Topics** folder, then click **Test Publication....**
The **Publish Test Message** application opens.
 - c) In the **Topic** field, type the name of the topic on which you want to publish the message.
You or another publisher can already be registered to publish on the topic, or you can enter a new topic string. When you publish the message, you are automatically registered as a publisher on the topic.
 - d) In the **Message data** field, type a message to send in the publication.
For example, type Hello, world!
 - e) Click **Publish message** to send the message to the Pub/Sub Engine.
The subscriber receives the message (the publication).
3. Start another instance of the **Subscribe** application.
The second **Subscribe** application does not receive the message that was published by the **Publish Test Message** application because it was not subscribing to the topic at the time that the publication was sent to the Publish/Subscribe Engine.
4. Unsubscribe the second **Subscribe** instance from the topic.
 - a) In the second **Subscribe** application, click **Unsubscribe**.
The second **Subscribe** application can no longer receive publications on that topic. The first **Subscribe** application can still receive publications on that topic.
5. Publish a retained publication to the topic.
 - a) In the **Publish Test Message** application, select the **Retained message** check box.
 - b) Change the text in the **Message data** field.
For example, type Hi, I'm home.
 - c) Click **Publish message**.
The retained publication is published to the Publish/Subscribe Engine. The first **Subscribe** application receives the retained publication. The second **Subscribe** application does not receive the publication because it is currently not subscribed.
6. Subscribe the second **Subscribe** application to the topic again:
 - a) In the second **Subscribe** application, click **Subscribe**.
The second **Subscribe** application is subscribed to the topic again and receives the retained publication because the Publish/Subscribe Engine kept a copy of the retained publication.

Results

You have now published and subscribed to test publications, including retained publications.

Related concepts

[“Publications” on page 33](#)

Publications are messages that are sent by an application to the Publish/Subscribe Engine. The Publish/Subscribe Engine then sends the messages on to any applications that have subscribed to receive the messages.

[“Topics” on page 30](#)

A topic identifies what a publication is about. A topic is a character string that describes the subject of the information that is published in a Publish/Subscribe message. As a subscriber, you can specify a topic or range of topics by using wildcards to receive the information that you require.

Sending and receiving test publications for specific topics

You can send (publish) and receive (subscribe to) test publications (messages) to check that your Publish/Subscribe Engine network and topics work as intended. You can configure a publication so that the Publish/Subscribe Engine retains a copy after publishing it to the subscribers. This enables new subscribers to receive the publication even if they subscribed after it was published.

Before you begin

Before you start:

- [Show the queue manager that hosts the Publish/Subscribe Engine.](#)

About this task

To send and receive test publications for a specific topic:

Procedure

1. Subscribe to the topic that you want to test:
 - a) In the Navigator view, expand the queue manager that hosts the Publish/Subscribe Engine.
 - b) Click the **Topics** folder.
All the topics display in the **Content** view.
 - c) Right-click a specific topic in the **Content** view, then click **Test Subscription....**
The **Subscribe** application opens.
2. Publish a message to the same topic:
 - a) In the Navigator view, expand the queue manager that hosts the Publish/Subscribe Engine.
 - b) Click the **Topics** folder.
All the topics display in the **Content** view.
 - c) Right-click the a specific topic in the **Content** view, then click **Test Publication....**
The **Publish Test Message** application opens.
 - d) In the **Message data** field, type a message to send in the publication.
For example, type `Hello, world!`
 - e) Click **Publish message** to send the message to the Publish/Subscribe Engine.
The subscriber receives the message (the publication).
3. Start another instance of the **Subscribe** application.
The second **Subscribe** application does not receive the message that was published by the **Publish Test Message** application because it was not subscribing to the topic at the time that the publication was sent to the Publish/Subscribe Engine.
4. Unsubscribe the second **Subscribe** instance from the topic.
 - a) In the second **Subscribe** application, click **Unsubscribe**.

The second **Subscribe** application can no longer receive publications on that topic. The first **Subscribe** application can still receive publications on that topic.

5. Publish a retained publication to the topic.

- a) In the **Publish Test Message** application, select the **Retained message** check box.
- b) Change the text in the **Message data** field.
For example, type `Hi, I'm home`.
- c) Click **Publish message**.

The retained publication is published to the Publish/Subscribe Engine. The first **Subscribe** application receives the retained publication. The second **Subscribe** application does not receive the publication because it is currently not subscribed.

6. Subscribe the second **Subscribe** application to the topic again:

- a) In the second **Subscribe** application, click **Subscribe**.

The second **Subscribe** application is subscribed to the topic again and receives the retained publication because the Publish/Subscribe Engine kept a copy of the retained publication.

Results

You have now published and subscribed to test publications, including retained publications on a specific topic.

Related concepts

[“Publications” on page 33](#)

Publications are messages that are sent by an application to the Publish/Subscribe Engine. The Publish/Subscribe Engine then sends the messages on to any applications that have subscribed to receive the messages.

[“Topics” on page 30](#)

A topic identifies what a publication is about. A topic is a character string that describes the subject of the information that is published in a Publish/Subscribe message. As a subscriber, you can specify a topic or range of topics by using wildcards to receive the information that you require.

Viewing topic status for publishers

Each topic can have many properties and values associated with it. When a topic has been assigned as a publisher, you can view its status, and edit the scheme to display the status information.

Before you begin

Before you start:

- [Show the queue manager that hosts the Publish/Subscribe Engine](#).

About this task

To view the status of a topic object publisher:

Procedure

1. In the **Navigator** view, expand the queue manager that hosts the Publish/Subscribe Engine, then click the **Topics** folder.
The existing topics on the Publish/Subscribe Engine are shown in the Content view.
2. In the **Content** view, right-click the topic that you want to view the publisher status for, then click **Topic Status - Publishers...**

Results

The **Status** dialog opens displaying the status of the topic object publisher.

What to do next

You can edit the way the information is presented in the **Status** dialog. For more information, see the following links.

Related concepts

[“Topics” on page 30](#)

A topic identifies what a publication is about. A topic is a character string that describes the subject of the information that is published in a Publish/Subscribe message. As a subscriber, you can specify a topic or range of topics by using wildcards to receive the information that you require.

[“Define schemes to change the order of columns in tables” on page 250](#)

When object data is displayed in IBM MQ Explorer in tables, you can customized the order of the columns in the tables.

Related tasks

[“Viewing topic status for subscribers” on page 136](#)

Each topic can have many properties and values associated with it. When a topic has been assigned as a subscriber, you can view its status, and edit the scheme to display the status information.

[“Creating a scheme” on page 251](#)

You can create schemes for most of the tables of data in IBM MQ Explorer.

[“Editing an existing scheme” on page 252](#)

You can edit any schemes that you have created previously and you can also edit the schemes that are supplied with IBM MQ Explorer, for example, the Standard for Queues scheme.

[“Copying an existing scheme” on page 253](#)

If there already exists a scheme that is similar to a scheme that you want to create, you can copy the existing scheme and then edit it as required.

[“Filtering the objects displayed in tables” on page 225](#)

When object data is displayed in IBM MQ Explorer in tables, you can filter the data so that only the objects in which you are interested are displayed.

Viewing topic status for subscribers

Each topic can have many properties and values associated with it. When a topic has been assigned as a subscriber, you can view its status, and edit the scheme to display the status information.

Before you begin

The queue manager that hosts the Publish/Subscribe engine must be visible in the **Navigator** view. To show the queue manager, follow the instructions in: [“Showing or hiding a queue manager” on page 98](#)

About this task

To view the status of a topic object subscriber:

Procedure

1. In the **Navigator** view, expand the queue manager that hosts the Publish/Subscribe Engine, then click the **Topics** folder.
The existing topics on the Publish/Subscribe Engine are shown in the **Content** view.
2. In the **Content** view, right-click the topic that you want to view the subscriber status for, then click **Topic Status - Subscribers**

Results

The **Status** dialog opens displaying the status of the topic object subscriber.

What to do next

You can edit the way the information is presented in the **Status** dialog. For more information, see the following links.

Related concepts

[“Topics” on page 30](#)

A topic identifies what a publication is about. A topic is a character string that describes the subject of the information that is published in a Publish/Subscribe message. As a subscriber, you can specify a topic or range of topics by using wildcards to receive the information that you require.

[“Define schemes to change the order of columns in tables” on page 250](#)

When object data is displayed in IBM MQ Explorer in tables, you can customized the order of the columns in the tables.

Related tasks

[“Viewing topic status for publishers” on page 135](#)

Each topic can have many properties and values associated with it. When a topic has been assigned as a publisher, you can view its status, and edit the scheme to display the status information.

[“Creating a scheme” on page 251](#)

You can create schemes for most of the tables of data in IBM MQ Explorer.

[“Editing an existing scheme” on page 252](#)

You can edit any schemes that you have created previously and you can also edit the schemes that are supplied with IBM MQ Explorer, for example, the Standard for Queues scheme.

[“Copying an existing scheme” on page 253](#)

If there already exists a scheme that is similar to a scheme that you want to create, you can copy the existing scheme and then edit it as required.

[“Filtering the objects displayed in tables” on page 225](#)

When object data is displayed in IBM MQ Explorer in tables, you can filter the data so that only the objects in which you are interested are displayed.

Creating a new subscription

You can create a new subscription to subscribe to a topic for an IBM WebSphere MQ 7.0, or later, queue manager.

About this task

To create a new subscription:

Procedure

1. In the **Navigators** view, expand the queue manager that you want to create a new subscription on.
2. Right-click the **Subscriptions** object-folder, then click **New > Subscription...**

Results

The **New Subscription** wizard opens. You can now work through the wizard to create a new subscription.

Related concepts

[“Topics” on page 30](#)

A topic identifies what a publication is about. A topic is a character string that describes the subject of the information that is published in a Publish/Subscribe message. As a subscriber, you can specify a topic or range of topics by using wildcards to receive the information that you require.

Related tasks

[“Configuring publish/subscribe for IBM WebSphere MQ 7.0 and later queue managers” on page 129](#)

In IBM MQ Explorer, you can configure IBM WebSphere MQ 7.0 and later queue managers as Publish/Subscribe Engines to route messages between publishing applications and subscribing applications. To test your configurations, you can register as a subscriber, and send and receive test publications if you are authorized to do so.

Related reference

[“IBM MQ Explorer Navigators view” on page 316](#)

The Navigator view in IBM MQ Explorer displays all of the IBM MQ objects that you can administer and monitor in IBM MQ Explorer.

Viewing a list of subscribers

You can view a list of applications that are subscribed to topics on a Publish/Subscribe Engine, or a list of applications that are subscribed to a specific topic.

About this task

To view a list of subscribers:

Procedure

In the **Navigator** view, expand the queue manager that hosts the Publish/Subscribe Engine which you want to view the subscribers of, then click the **Subscriptions** object-folder.

Results

The existing subscriptions on the Publish/Subscribe Engine are shown in the **Content** view.

Related concepts

[“Topics” on page 30](#)

A topic identifies what a publication is about. A topic is a character string that describes the subject of the information that is published in a Publish/Subscribe message. As a subscriber, you can specify a topic or range of topics by using wildcards to receive the information that you require.

Related reference

[“IBM MQ Explorer Content view” on page 324](#)

The Content view in IBM MQ Explorer displays information about objects and properties.

[“IBM MQ Explorer Navigator view” on page 316](#)

The Navigator view in IBM MQ Explorer displays all of the IBM MQ objects that you can administer and monitor in IBM MQ Explorer.

Refreshing proxy subscriptions

You can refresh proxy subscriptions on an IBM MQ queue manager.

About this task

Refreshing proxy subscriptions resynchronizes all the proxy subscriptions with all other directly connected queue managers in any cluster or hierarchy in which this queue manager is participating. You must refresh the proxy subscriptions only in exceptional circumstances, for example, when the queue manager is receiving subscriptions that it must not be sent, or not receiving subscriptions that it must receive. The following list describes some of the exceptional reasons for refreshing proxy subscriptions:

- Disaster recovery.
- Problems that are identified in a queue manager error log where messages inform of the issuing of the REFRESH QMGR TYPE(REPOS) command.
- Operator errors, for example, issuing a DELETE SUB command on a proxy subscription.

Missing proxy subscriptions can be caused if the closest matching topic definition is specified with **Subscription scope** set to Queue Manager or it has an empty or incorrect cluster name. Note that **Publication scope** does not prevent the sending of proxy subscriptions, but does prevent publications from being delivered to them.

Extraneous proxy subscriptions can be caused if the closest matching topic definition is specified with **Proxy subscription behavior** set to Force.

Missing or extraneous proxy subscriptions that are due to configuration errors are not changed by issuing a resynchronization. A resynchronization does resolve missing or extraneous publications as a result of the exceptional reasons listed.

To refresh the proxy subscriptions of a queue manager:

Procedure

1. In the **Navigator** view, select the queue manager that you want refresh the proxy subscriptions of.
2. Right-click the queue manager, then click **Publish/Subscribe > Refresh Proxy Subscriptions**.

Results

The **Refresh proxy subscriptions** dialog opens. You can now click **Yes** to refresh the proxy subscriptions, or click **No** to close the dialog.

Related concepts

[“Subscriptions” on page 33](#)

A subscription is a record that contains the information about the topic or topics that the subscriber is interested in and wants to receive information about. Thus, the subscription information determines which publications get forwarded to the subscriber. Subscribers can receive information from many different publishers, and the information they receive can also be sent to other subscribers.

[“Topics” on page 30](#)

A topic identifies what a publication is about. A topic is a character string that describes the subject of the information that is published in a Publish/Subscribe message. As a subscriber, you can specify a topic or range of topics by using wildcards to receive the information that you require.

Related tasks

[“Configuring publish/subscribe for IBM WebSphere MQ 7.0 and later queue managers” on page 129](#)

In IBM MQ Explorer, you can configure IBM WebSphere MQ 7.0 and later queue managers as Publish/Subscribe Engines to route messages between publishing applications and subscribing applications. To test your configurations, you can register as a subscriber, and send and receive test publications if you are authorized to do so.

Related reference

[“IBM MQ Explorer Navigator view” on page 316](#)

The Navigator view in IBM MQ Explorer displays all of the IBM MQ objects that you can administer and monitor in IBM MQ Explorer.

Creating a new Multicast communication information object

IBM MQ Multicast offers low latency, high fan out, reliable multicast messaging.

About this task

Multicast is more efficient than traditional unicast publish/subscribe messaging, and can be scaled to a high number of subscribers. IBM MQ enables reliable Multicast messaging by using acknowledgments, negative acknowledgments, and sequence numbers to achieve low latency messaging with high fan out.

IBM MQ Multicast's fair delivery enables near simultaneous delivery, ensuring that no recipient gains an advantage. As IBM MQ Multicast uses the network to deliver messages, a publish/subscribe engine is not needed to fan-out data. After a topic is mapped to a group address, there is no need for a queue manager as publishers and subscribers can operate in a peer-to-peer mode. This allows the load to be reduced on queue manager servers, and the queue manager server is no longer a potential point of failure.

To create a new topic in the IBM MQ Explorer:

Procedure

1. Expand the queue manager that you want to host the multicast communication information object on to display the object-folders in the **Navigator** view.
2. Right-click **Communication Information**, then click **New > Multicast Communication Information**.

Results

The **Communication Information** wizard opens. Work through the wizard to create a new communication information object.

Related reference

[“Multicast Communication Information object properties” on page 505](#)

You can set attributes for Multicast communication information objects.

Managing multi-instance queue managers

You must configure the IBM MQ Explorer to manage multi-instance queue managers using remote connections.

Use the **Queue Managers > Add Remote Queue Manager** menu item to add connections to a multi-instance queue manager. Alternatively, if you have already configured a remote connection to a queue manager, right-click the remote queue manager node in the IBM MQ Explorer navigator, then click **Connection Details > Manage Instances** to add, remove, test, and reorder connections.

You must connect all instances of a queue manager to its remote queue manager node, and then the node is able to monitor the status of all the instances of the queue manager. You can see which instance is currently active, and which instances are in standby or disconnected.

It is important that you test the connections to all the instances of the queue manager when they are in standby and in active status. Ensure that the listener is running on both the active and the standby instances of the queue manager. A standby-instance queue manager does not have access to the queue manager file system, and does not automatically start listeners until it becomes active. To test the connections of both active and standby instances, consider starting the listener for both instances from the command line.

Do not start listeners manually when they are configured to start with the queue manager. This produces errors when the queue manager starts because the listener service fails due to the port already being in use.

Connecting directly to a multi-instance queue manager

Create direct remote connections to multiple instances of a queue manager to administer a multi-instance queue manager using IBM MQ Explorer.

On the **Specify new connection details** page, you are asked to provide connection information for two instances of the queue manager. Both these instances might be remote, or one might be local and one remote. The IBM MQ Explorer creates a single remote queue manager node in the navigation tree to represent both instances of the queue manager. You can see the overall status of the multi-instance queue manager.

Once you have already created a remote queue manager node in the IBM MQ Explorer, you can use it to add and remove additional instances of the queue manager. You cannot add additional queue manager instances to a local queue manager node.

Before connecting to a multi-instance queue manager you need to create a multi-instance queue manager.

Related concepts

[Creating a multi-instance queue manager](#)

You cannot create all the instances of a multi-instance queue manager using the IBM MQ Explorer.

[Deleting a multi-instance queue manager](#)

The IBM MQ Explorer does not provide a means to delete all the instances of a multi-instance queue manager.

[Starting a multi-instance queue manager](#)

You can start a multi-instance queue manager in two ways from a local queue manager in the IBM MQ Explorer.

[Stopping a multi-instance queue manager](#)

You can stop a multi-instance queue manager in two ways from a local queue manager in the IBM MQ Explorer.

Managing connections to multi-instance queue managers

You can manage the connections IBM MQ Explorer uses to connect to a queue manager for remote administration. You need multiple connections to monitor the status of all the instances of a multi-instance queue manager. You can also configure multiple connections to a single instance of a queue manager to make remote administration more reliable.

Creating a multi-instance queue manager

You cannot create all the instances of a multi-instance queue manager using the IBM MQ Explorer.

To create a highly available multi-instance queue manager, first create a queue manager on one server with shared data and log directories on a highly available network storage device on a different server, and then add the queue manager definition stanza to the `mqs.ini` file on another server of the same architecture, and running the same or later version of IBM MQ. The commands **dspmqinf** and **addmqinf** help you copy the queue manager definition from the first to the second server without having to edit the `mqs.ini` file manually.

For more information, see [Multi-instance queue managers](#) in the IBM MQ online product documentation.

Related concepts

Connecting directly to a multi-instance queue manager

Create direct remote connections to multiple instances of a queue manager to administer a multi-instance queue manager using IBM MQ Explorer.

Deleting a multi-instance queue manager

The IBM MQ Explorer does not provide a means to delete all the instances of a multi-instance queue manager.

Starting a multi-instance queue manager

You can start a multi-instance queue manager in two ways from a local queue manager in the IBM MQ Explorer.

Stopping a multi-instance queue manager

You can stop a multi-instance queue manager in two ways from a local queue manager in the IBM MQ Explorer.

Managing connections to multi-instance queue managers

You can manage the connections IBM MQ Explorer uses to connect to a queue manager for remote administration. You need multiple connections to monitor the status of all the instances of a multi-instance queue manager. You can also configure multiple connections to a single instance of a queue manager to make remote administration more reliable.

Deleting a multi-instance queue manager

The IBM MQ Explorer does not provide a means to delete all the instances of a multi-instance queue manager.

To delete a multi-instance queue manager, you should delete the queue manager from one server, and then use the **rmvmqinf** command to remove the queue manager definitions from other servers.

For more information, see [Multi-instance queue managers](#) in the IBM MQ online product documentation.

Note: If you delete the queue manager again, but on another server that has the same queue manager defined, the **dltmqm** command fails. If you do try to delete a queue manager on a server that had a queue manager definition, but no queue manager, delete the queue manager again on the same server, and the queue manager is completely removed.

Related concepts

Connecting directly to a multi-instance queue manager

Create direct remote connections to multiple instances of a queue manager to administer a multi-instance queue manager using IBM MQ Explorer.

Creating a multi-instance queue manager

You cannot create all the instances of a multi-instance queue manager using the IBM MQ Explorer.

Starting a multi-instance queue manager

You can start a multi-instance queue manager in two ways from a local queue manager in the IBM MQ Explorer.

Stopping a multi-instance queue manager

You can stop a multi-instance queue manager in two ways from a local queue manager in the IBM MQ Explorer.

Managing connections to multi-instance queue managers

You can manage the connections IBM MQ Explorer uses to connect to a queue manager for remote administration. You need multiple connections to monitor the status of all the instances of a multi-instance queue manager. You can also configure multiple connections to a single instance of a queue manager to make remote administration more reliable.

Starting a multi-instance queue manager

You can start a multi-instance queue manager in two ways from a local queue manager in the IBM MQ Explorer.

As a single-instance queue manager

Do not check the **Permit a standby instance** check box.

As a multi-instance queue manager

Start the first instance, checking the **Permit a standby instance** check box, then start the second instance, also checking the **Permit a standby instance** check box.

Note: You cannot use a remotely connected queue manager to start a multi-instance queue manager.

Related concepts

Connecting directly to a multi-instance queue manager

Create direct remote connections to multiple instances of a queue manager to administer a multi-instance queue manager using IBM MQ Explorer.

Creating a multi-instance queue manager

You cannot create all the instances of a multi-instance queue manager using the IBM MQ Explorer.

Deleting a multi-instance queue manager

The IBM MQ Explorer does not provide a means to delete all the instances of a multi-instance queue manager.

Stopping a multi-instance queue manager

You can stop a multi-instance queue manager in two ways from a local queue manager in the IBM MQ Explorer.

Managing connections to multi-instance queue managers

You can manage the connections IBM MQ Explorer uses to connect to a queue manager for remote administration. You need multiple connections to monitor the status of all the instances of a multi-instance queue manager. You can also configure multiple connections to a single instance of a queue manager to make remote administration more reliable.

Stopping a multi-instance queue manager

You can stop a multi-instance queue manager in two ways from a local queue manager in the IBM MQ Explorer.

Stop all instances of the queue manager

Do not check the **Allow switchover to standby instance** check box.

Stop this instance of the queue manager, switching over to a standby instance

Check the **Allow switchover to standby instance** check box. If there is no standby instance running the command fails and leaves the queue manager running.

Note: You cannot use a remotely connected queue manager to stop a multi-instance queue manager.

Related concepts

[Connecting directly to a multi-instance queue manager](#)

Create direct remote connections to multiple instances of a queue manager to administer a multi-instance queue manager using IBM MQ Explorer.

[Creating a multi-instance queue manager](#)

You cannot create all the instances of a multi-instance queue manager using the IBM MQ Explorer.

[Deleting a multi-instance queue manager](#)

The IBM MQ Explorer does not provide a means to delete all the instances of a multi-instance queue manager.

[Starting a multi-instance queue manager](#)

You can start a multi-instance queue manager in two ways from a local queue manager in the IBM MQ Explorer.

[Managing connections to multi-instance queue managers](#)

You can manage the connections IBM MQ Explorer uses to connect to a queue manager for remote administration. You need multiple connections to monitor the status of all the instances of a multi-instance queue manager. You can also configure multiple connections to a single instance of a queue manager to make remote administration more reliable.

Managing connections to multi-instance queue managers

You can manage the connections IBM MQ Explorer uses to connect to a queue manager for remote administration. You need multiple connections to monitor the status of all the instances of a multi-instance queue manager. You can also configure multiple connections to a single instance of a queue manager to make remote administration more reliable.

The queue manager instances must share the same queue manager data; either by configuring multiple connections to the same queue manager on a single server, or by configuring connections to multiple instances of the same queue manager on different servers.

You cannot remove the active connection, which IBM MQ Explorer is using.

Click **Test connections** to refresh the status of the connections.

To connect to a standby queue manager instance you must have configured a listener process to be running while the queue manager is in standby status. For example, set the listener CONTROL to Queue Manager or Queue Manager Start.

Related concepts

[Connecting directly to a multi-instance queue manager](#)

Create direct remote connections to multiple instances of a queue manager to administer a multi-instance queue manager using IBM MQ Explorer.

[Creating a multi-instance queue manager](#)

You cannot create all the instances of a multi-instance queue manager using the IBM MQ Explorer.

[Deleting a multi-instance queue manager](#)

The IBM MQ Explorer does not provide a means to delete all the instances of a multi-instance queue manager.

[Starting a multi-instance queue manager](#)

You can start a multi-instance queue manager in two ways from a local queue manager in the IBM MQ Explorer.

[Stopping a multi-instance queue manager](#)

You can stop a multi-instance queue manager in two ways from a local queue manager in the IBM MQ Explorer.

Creating and configuring a queue manager cluster

A cluster is a group of two or more queue managers that are logically associated and can share information with each other. You can use the wizards and properties dialogs in IBM MQ Explorer to create and configure queue manager clusters.

About this task

Since the queue managers in a cluster are logically associated and can share information with each other, this means that an application can put a message on a cluster queue from any queue manager in the cluster and the message is automatically routed to the queue manager where the cluster queue is defined. The amount of system administration is reduced because the cluster channels that the cluster queue managers use to exchange application messages are automatically defined as required.

IBM MQ Explorer provides wizards to help you to create and configure queue manager clusters and objects.

You cannot use the wizards to manage queue managers and objects that belong to more than one cluster (and, therefore, use namelists). However, you can still use the IBM MQ Explorer properties dialogs to edit the properties of the queue managers and objects if you want them to belong to more than one cluster.

The following topics describe how to create and configure queue manager clusters in IBM MQ Explorer:

- [“Creating a queue manager cluster” on page 145](#)
- [“Adding a queue manager to a cluster” on page 146](#)
- [“Removing a queue manager from a cluster” on page 147](#)
- [“Suspending the cluster membership of a queue manager” on page 148](#)
- [“Resuming the cluster membership of a queue manager” on page 148](#)
- [“Refreshing locally held information about a cluster” on page 149](#)
- [“Specifying a different cluster information source for IBM MQ Explorer ” on page 150](#)
- [“Cluster repositories” on page 151](#)
- [“Making a queue manager a full repository for more than one cluster” on page 151](#)
- [“Sharing a queue in a cluster” on page 152](#)
- [“Connecting to a remote cluster queue manager” on page 153](#)
- [“Administering a remote cluster queue manager” on page 154](#)

For more information, see [Distributed queuing and clusters](#) in the IBM MQ online product documentation.

Related concepts

[“Queue manager clusters” on page 48](#)

A cluster is a group of two or more queue managers that are logically associated and can share information with each other. Any queue manager can send a message to any other queue manager in the same cluster without you needing to set up a specific channel definition, remote queue definition, or transmission queue, because all of this information is held in the repository, to which all queue managers in the cluster have access.

Creating a queue manager cluster

IBM MQ Explorer treats queue manager clusters as objects so that you can create and administer them like other MQ objects.

About this task

All the queue manager clusters that are known to IBM MQ Explorer are displayed in the **Queue Manager Clusters** folder.

Before you can create a new queue manager cluster:

- Create two queue managers that will have the full repositories for the cluster.
- Each full repository queue manager in the cluster must have a running listener.
- You must know the connection details of each full repository queue manager in the cluster because you are asked to enter these details in the wizard.

Note: You cannot use the Create Cluster wizard if the full repository queue managers already belong to another cluster. If you want to use queue managers that already belong to another cluster, you must configure the cluster using the MQSC commands.

To create a new cluster, complete the following steps.

Procedure

1. In the Navigator view, right-click the **Queue Manager Clusters** folder, then click **New...** The Create Cluster wizard opens.
2. Work through the pages in the wizard to enter the following information about the new cluster:
 - a) Page 1: The name of the new cluster. This name must be unique in your organization.
 - b) Page 2: The name of a queue manager that will have a full repository of information about the cluster. The queue manager must already exist; click **Add queue manager to MQ Explorer** if the queue manager is not already known to IBM MQ Explorer.
 - c) Page 3: The name of a second queue manager that will have a full repository of information about the cluster. The queue manager must already exist; click **Add queue manager to MQ Explorer** if the queue manager is not already known to IBM MQ Explorer.
 - d) Page 4: The connection name of the first full repository queue manager. The format of the connection name depends on the transport protocol that the queue manager uses. For example, if the queue manager uses TCP/IP, you can use the format *computer_name(port_number)* where *computer_name* is the name of the computer that hosts the queue manager, and *port_number* is the port number on which the queue manager listens for connections.
3. Click **Finish** to create the cluster.

Results

The new cluster is displayed in the **Queue Manager Clusters** folder. The cluster's full repositories are shown in its **Full Repositories** folder.

For more information, see [Distributed queuing and clusters](#) and [Script \(MQSC\) Commands](#) in the IBM MQ online product documentation.

Related concepts

[“Queue manager clusters” on page 48](#)

A cluster is a group of two or more queue managers that are logically associated and can share information with each other. Any queue manager can send a message to any other queue manager in the same cluster without you needing to set up a specific channel definition, remote queue definition, or transmission queue, because all of this information is held in the repository, to which all queue managers in the cluster have access.

[“Cluster repositories” on page 151](#)

A cluster repository contains information about the cluster; for example, information about the queue managers that are members of the cluster, and the cluster channels. Repositories are hosted by the queue managers in the cluster.

Related tasks

[“Adding a queue manager to a cluster” on page 146](#)

You can add a queue manager to a cluster as either a full repository or a partial repository.

Adding a queue manager to a cluster

You can add a queue manager to a cluster as either a full repository or a partial repository.

About this task

This task shows how to add a queue manager to an existing cluster using the Create Cluster wizard in IBM MQ Explorer. You can use the wizard to add a queue manager, provided that the queue manager does not already belong to another cluster.

Although you can add any queue manager to an existing cluster, even if the queue manager already belongs to another cluster, you cannot use the Create Cluster wizard if the queue manager already belongs to another cluster. You must configure the cluster using MQSC commands.

Before you can add a queue manager to a cluster:

- Create the queue manager.
- The queue manager must have a running listener.
- You must know the connection details of the queue manager because the you are asked to enter these details in the wizard.

To add a queue manager to a cluster:

Procedure

1. In the Navigator view, right-click the cluster, then click **Add queue manager to cluster**. The Create Cluster wizard opens.
2. Work through the pages in the wizard to enter the following information about the queue manager:
 - a) Page 1: The name of the queue manager. The queue manager must already exist; click **Add queue manager to MQ Explorer** if the queue manager is not already known to IBM MQ Explorer.
 - b) Page 2: Whether the queue manager will be a full repository or a partial repository for the cluster.
 - c) Page 3: The connection name of the queue manager. The format of the connection name depends on the transport protocol that the queue manager uses. For example, if the queue manager uses TCP/IP, you can use the format *computer_name(port_number)* where *computer_name* is the name or IP address of the computer that hosts the queue manager, and *port_number* is the port number on which the queue manager listens for connections.
 - d) Page 4: If the queue manager will be a partial repository, select one or more full repository queue managers to which the partial repository queue manager will send information about the cluster.
 - e) Page 5: If the queue manager will be a partial repository, select the cluster-receiver channel that the full repository queue manager will use to receive information from the partial repository queue manager.
 - f) Page 6: Using the list, specify a cluster-sender channel on each full repository queue manager that they will use to send cluster information to the new repository queue manager.
3. Click **Finish** to add the queue manager the cluster.

Results

The queue manager is added to the cluster as a full repository or a partial repository. The queue manager is displayed in the **Full Repository** folder or the **Partial Repository** folder for the cluster.

For more information, see [Distributed queuing and clusters](#) and [Script \(MQSC\) Commands](#) in the IBM MQ online product documentation.

Related concepts

[“Queue manager clusters” on page 48](#)

A cluster is a group of two or more queue managers that are logically associated and can share information with each other. Any queue manager can send a message to any other queue manager in the same cluster without you needing to set up a specific channel definition, remote queue definition, or transmission queue, because all of this information is held in the repository, to which all queue managers in the cluster have access.

[“Cluster repositories” on page 151](#)

A cluster repository contains information about the cluster; for example, information about the queue managers that are members of the cluster, and the cluster channels. Repositories are hosted by the queue managers in the cluster.

Related tasks

[“Creating a queue manager cluster” on page 145](#)

IBM MQ Explorer treats queue manager clusters as objects so that you can create and administer them like other MQ objects.

Removing a queue manager from a cluster

If you no longer want a queue manager to be a member of a cluster, you can remove the queue manager from the cluster.

About this task

When you remove the queue manager from the cluster using IBM MQ Explorer, the queue manager's properties are updated; the table on the Cluster page of the queue manager's properties dialog is updated, and if the queue manager was a full repository for the cluster, the attributes on the Repository page of the queue manager's properties dialog are also updated.

When you remove a queue manager from a cluster, the queue manager's cluster queues and cluster channels are no longer available to applications using the cluster.

Note that if the queue manager belongs to more than one cluster (using namelists), you cannot remove a queue manager from a cluster using the following instructions; you must manually edit the queue manager's properties.

To remove a queue manager from a cluster:

Procedure

1. In the Navigator view (in the **Queue Manager Clusters** folder), expand the cluster from which the queue manager is currently suspended.
2. Right-click the queue manager, then click **Remove queue manager from cluster...**
3. When you are prompted, click **Yes**.

Results

The queue manager is removed from the cluster and the queue manager's properties are updated.

Related tasks

[“Suspending the cluster membership of a queue manager” on page 148](#)

If a queue manager is a member of a cluster but you want to temporarily prevent the queue manager sharing its cluster queues and exchanging messages using the cluster, you can suspend the queue manager from the cluster. You can later easily resume the queue manager's membership of the cluster.

[“Adding a queue manager to a cluster” on page 146](#)

You can add a queue manager to a cluster as either a full repository or a partial repository.

[“Creating and configuring a queue manager cluster” on page 144](#)

A cluster is a group of two or more queue managers that are logically associated and can share information with each other. You can use the wizards and properties dialogs in IBM MQ Explorer to create and configure queue manager clusters.

[Removing a queue manager from a cluster: best practice](#)

[Removing a queue manager from a cluster: alternative method](#)

Suspending the cluster membership of a queue manager

If a queue manager is a member of a cluster but you want to temporarily prevent the queue manager sharing its cluster queues and exchanging messages using the cluster, you can suspend the queue manager from the cluster. You can later easily resume the queue manager's membership of the cluster.

About this task

To suspend a queue manager without using IBM MQ Explorer, see [Suspend Cluster Queue Manager \(SPDMQMCLQM\)](#) in the IBM MQ online product documentation.

To suspend a queue manager from a cluster, in the Navigator view (in the **Queue Manager Clusters** folder), right-click the queue manager, and then click **Suspend cluster membership....**

The queue manager is suspended from the cluster and its icon is decorated to show this.

For more information, see [Distributed queuing and clusters](#) in the IBM MQ online product documentation.

Related tasks

[“Resuming the cluster membership of a queue manager” on page 148](#)

If you previously suspended a queue manager's membership of a cluster, but then want to resume the queue manager's membership, you can do this without having to re-enter the queue manager's connection details.

[“Removing a queue manager from a cluster” on page 147](#)

If you no longer want a queue manager to be a member of a cluster, you can remove the queue manager from the cluster.

[“Creating and configuring a queue manager cluster” on page 144](#)

A cluster is a group of two or more queue managers that are logically associated and can share information with each other. You can use the wizards and properties dialogs in IBM MQ Explorer to create and configure queue manager clusters.

Resuming the cluster membership of a queue manager

If you previously suspended a queue manager's membership of a cluster, but then want to resume the queue manager's membership, you can do this without having to re-enter the queue manager's connection details.

About this task

If you have suspended the cluster membership of a queue manager, the queue manager is unable to exchange messages using the cluster, and the queue manager's cluster queues are unavailable to the other queue managers in the cluster. You can easily resume the queue manager's cluster membership without having to enter the queue manager's connection details again.

For more information, see [Distributed queuing and clusters](#) in the IBM MQ online product documentation.

To resume the cluster membership of a queue manager:

Procedure

In the Navigator view (in the **Queue Manager Clusters** folder), right-click the queue manager, then click **Resume cluster membership...**

Results

The queue manager is an active member of the cluster again and any decoration is removed from the queue manager's icon to show this.

Related tasks

[“Suspending the cluster membership of a queue manager” on page 148](#)

If a queue manager is a member of a cluster but you want to temporarily prevent the queue manager sharing its cluster queues and exchanging messages using the cluster, you can suspend the queue manager from the cluster. You can later easily resume the queue manager's membership of the cluster.

[“Creating and configuring a queue manager cluster” on page 144](#)

A cluster is a group of two or more queue managers that are logically associated and can share information with each other. You can use the wizards and properties dialogs in IBM MQ Explorer to create and configure queue manager clusters.

Refreshing locally held information about a cluster

You are unlikely to need to refresh (discard) all locally held information about a cluster in normal circumstances but you might be asked to do this by your IBM Support Center.

Before you begin

For large clusters, use of the **REFRESH CLUSTER** command can be disruptive to the cluster while it is in progress, and again at 27 day intervals thereafter when the cluster objects automatically send status updates to all interested queue managers. See [Clustering: Using REFRESH CLUSTER best practices](#) in the online product documentation.

Procedure

1. In the Navigator view (in the **Queue Manager Clusters** folder), right-click the queue manager, then click **Refresh cluster membership...** The Refresh Cluster Queue Managers dialog opens.
2. Select the scope of the refresh:

- To refresh all of the queue manager's information about the cluster except for the following information, click **Refresh cluster**:
 - The queue manager's knowledge of all cluster queue managers and cluster queues that are locally defined is retained.
 - The queue manager's knowledge of all cluster queue managers that are full repositories is retained.
 - If the queue manager is a full repository, knowledge of the other cluster queue managers in the cluster is retained. Everything else is removed from the local copy of the repository and is rebuilt from the other full repositories in the cluster.

In addition, to specify that objects representing full repository cluster queue managers are also refreshed, select **Clear repository information**. This option is available only to partial repository queue managers. You can, however, temporarily configure a full repository to be a partial repository so that you can refresh its repository too.

- To refresh the queue manager in all of the clusters to which it belongs, click **Refresh all clusters**.

In addition, to force the queue manager to restart its search for full repositories from the information in the local cluster-sender channel definitions, even if the cluster-sender channel connects the queue manager to several clusters, select **Clear repository information**.

3. Click **OK**.

Results

The queue manager's information about the cluster, or clusters, is refreshed.

For more information, see [Distributed queuing and clusters](#) in the IBM MQ online product documentation.

Related tasks

[“Creating and configuring a queue manager cluster” on page 144](#)

A cluster is a group of two or more queue managers that are logically associated and can share information with each other. You can use the wizards and properties dialogs in IBM MQ Explorer to create and configure queue manager clusters.

Related information

[Clustering: Using REFRESH CLUSTER best practices](#)

Specifying a different cluster information source for IBM MQ Explorer

You can change the full repository queue manager from which IBM MQ Explorer obtains information about which queue managers belong to a cluster.

About this task

For each cluster, IBM MQ Explorer obtains information from one of the cluster's full repository queue managers about which queue managers belong to the cluster. You can change the IBM MQ Explorer information source by specifying a different full repository queue manager that belongs to the same cluster.

To specify a different full repository queue manager, complete the following steps.

Procedure

1. In the Navigator view, click the cluster. The Content view displays the name of the full repository queue manager that is currently the information source.
2. In the Content view, click **Select...** A dialog opens.
3. From the list, select a full repository queue manager, then click **Finish**.

Results

The Content view now displays the name of the queue manager that you selected. IBM MQ Explorer updates its information about the cluster from the full repository queue manager specified.

For more information, see [Distributed queuing and clusters](#) in the IBM MQ online product documentation.

Related concepts

[“Cluster repositories” on page 151](#)

A cluster repository contains information about the cluster; for example, information about the queue managers that are members of the cluster, and the cluster channels. Repositories are hosted by the queue managers in the cluster.

Related tasks

[“Creating and configuring a queue manager cluster” on page 144](#)

A cluster is a group of two or more queue managers that are logically associated and can share information with each other. You can use the wizards and properties dialogs in IBM MQ Explorer to create and configure queue manager clusters.

Cluster repositories

A cluster repository contains information about the cluster; for example, information about the queue managers that are members of the cluster, and the cluster channels. Repositories are hosted by the queue managers in the cluster.

Normally, to ensure availability, two queue managers (on different computers) host full repositories, which contain a complete set of information about the cluster and its resources. The two queue managers exchange messages to keep their repositories synchronized. All the other queue managers in the cluster host partial repositories, which contain an incomplete set of information about the cluster and its resources.

A queue manager's partial repository contains only information about the queue managers with which the queue manager needs to exchange messages. The queue manager requests updates from the full repositories so that if the information changes, the full repository queue managers sends them the new information. For much of the time a queue manager's partial repository has all the information it needs to perform within the cluster. When a queue manager needs some additional information, it makes inquiries of the full repository and updates its partial repository.

Two special types of channel are used by each queue manager for this purpose, one each of cluster-sender (CLUSDR) and cluster-receiver (CLUSRCVR).

DHCP

If a computer uses DHCP (dynamic allocation of IP address), you are recommended to define the repository's `Connection` name attribute using the computer's name instead of the computer's IP address. This is because the connection name is used to find the repository. If the computer's IP address is used and the IP address subsequently changes, other queue managers will no longer be able to find the repository. This still applies even if all the queue managers in the cluster are on the same computer, because the IP address is still used to find the repository.

Related concepts

[“Queue manager clusters” on page 48](#)

A cluster is a group of two or more queue managers that are logically associated and can share information with each other. Any queue manager can send a message to any other queue manager in the same cluster without you needing to set up a specific channel definition, remote queue definition, or transmission queue, because all of this information is held in the repository, to which all queue managers in the cluster have access.

[“Channels” on page 34](#)

IBM MQ can use three different types of channels: a message channel, an MQI channel, and an AMQP channel.

Making a queue manager a full repository for more than one cluster

A queue manager can be a full repository for more than one cluster at the same time.

About this task

If you want a queue manager to be a full repository for more than one cluster, you must create a namelist for the queue manager and list the names of the clusters in the namelist. The **Create Cluster** wizard does not edit namelists so you must manage multiple clusters manually in IBM MQ Explorer.

To make a queue manager a full repository for more than one cluster, complete the following steps.

Procedure

1. Create a new namelist for the queue manager.
2. Open the new namelist's **Properties** dialog and edit the namelist:
 - a) On the **General** page of the **Properties** dialog, in the **Names** field, click **Edit**. The **Edit Names** dialog opens.
 - b) Click **Add**. The Add to Names dialog opens.
 - c) In the **Add to Names** dialog, type the name of a cluster for which you want the queue manager to be a full repository, then click **OK**.
 - d) Add the name of each cluster for which you want the queue manager to be a full repository.
 - e) In the **Edit Names** dialog, click **OK** to return to the **Properties** dialog.
 - f) Click **OK** to apply the changes and close the **Properties** dialog.
3. Open the queue manager's **Properties** dialog and specify the namelist:
 - a) On the **Repository** page of the **Properties** dialog, click **Repository for a list a clusters**, then type the name of the namelist in the field.
 - b) Click **OK** to apply the changes and close the **Properties** dialog.

Results

The queue manager is added to the **Full Repository** folder of the clusters that are listed in the namelist. Any of the clusters that were not previously shown in the **Queue Manager Clusters** folder are shown now.

Related concepts

[“Namelists” on page 39](#)

A namelist is an IBM MQ object that contains a list of names of other objects.

Related tasks

[“Creating and configuring queue managers and objects” on page 27](#)

You can create, configure, and delete queue managers and objects in IBM MQ Explorer by using the Navigator view and Content view.

[“Configuring queue managers and objects” on page 51](#)

You can configure many of the attributes of queue managers and their objects from IBM MQ Explorer using the properties dialogs.

Sharing a queue in a cluster

A queue manager that belongs to a cluster can share one or more of its queues with all the other members of the cluster.

About this task

Before you can share a queue in a cluster:

- The queue manager that owns the queue must be a member of the cluster.
- The queue manager's membership of the cluster must not be suspended.

To share a queue in a cluster, complete the following steps.

Procedure

1. In the Navigator view, click the queue manager's **Queues** folder. The queue manager's queues are displayed in the Content view.
2. In the Content view, right-click the queue that you want to share, then click **Properties...** The queue's Properties dialog opens.
3. On the **Cluster** page of the Properties dialog, click **Shared in cluster**, then type the name of the cluster in which you want to share the queue. If the queue is already shared in a cluster or if you want to

share the queue in more than one cluster, click **Shared in a list of clusters**, then type the name of the namelist that contains the list of clusters.

4. Click **OK** to apply the changes.

Results

The queue is now available to all the queue managers in the cluster or clusters in which the queue is shared.

Related concepts

[“Namelists” on page 39](#)

A namelist is an IBM MQ object that contains a list of names of other objects.

[“Queue manager clusters” on page 48](#)

A cluster is a group of two or more queue managers that are logically associated and can share information with each other. Any queue manager can send a message to any other queue manager in the same cluster without you needing to set up a specific channel definition, remote queue definition, or transmission queue, because all of this information is held in the repository, to which all queue managers in the cluster have access.

Related tasks

[“Creating a queue manager cluster” on page 145](#)

IBM MQ Explorer treats queue manager clusters as objects so that you can create and administer them like other MQ objects.

[“Adding a queue manager to a cluster” on page 146](#)

You can add a queue manager to a cluster as either a full repository or a partial repository.

[“Resuming the cluster membership of a queue manager” on page 148](#)

If you previously suspended a queue manager's membership of a cluster, but then want to resume the queue manager's membership, you can do this without having to re-enter the queue manager's connection details.

Connecting to a remote cluster queue manager

You can connect IBM MQ Explorer to a remote queue manager by using the cluster information source as an intermediary queue manager.

About this task

If a remote queue manager belongs to a cluster that is displayed in IBM MQ Explorer but IBM MQ Explorer does not know anything about the queue manager, the queue manager's icon shows that it is not connected. For IBM MQ Explorer to get information about the remote queue manager, it must connect to the queue manager. Of course, if you don't know the connection details of the queue manager, you cannot easily add it to the **Queue Managers** folder, and you might not want to be able to administer the queue manager anyway. Therefore, you can connect IBM MQ Explorer to the remote queue manager using the cluster information source as an intermediary queue manager.

For example, if QMX is the full repository queue manager from which IBM MQ Explorer obtains all its information about the cluster, you can connect to QMZ, the remote cluster queue manager, using QMX as an intermediary queue manager. This means that IBM MQ Explorer does not need to know the connection details of the remote cluster queue manager because QMX, the cluster's full repository queue manager, already has this information.

When IBM MQ Explorer is connected to the remote cluster queue manager, if you want to administer the remote cluster queue manager, you can then show the queue manager in the **Queue Managers** folder.

To connect to a remote cluster queue manager, in the Navigator view (in the **Queue Manager Clusters** folder), right-click the queue manager, and then click **Connect to queue manager**.

IBM MQ Explorer connects to the remote cluster queue manager using the full repository queue manager that is IBM MQ Explorer's information source about the cluster. Click the queue manager to display its cluster queues and cluster channels in the Content view.

For more information, see [Distributed queuing and clusters](#) in the IBM MQ online product documentation.

Related tasks

[“Creating and configuring a queue manager cluster” on page 144](#)

A cluster is a group of two or more queue managers that are logically associated and can share information with each other. You can use the wizards and properties dialogs in IBM MQ Explorer to create and configure queue manager clusters.

Administering a remote cluster queue manager

After connecting to a remote cluster queue manager using the cluster information source as an intermediate queue manager, you can select to show the queue manager in the **Queue Managers** folder. You can then use the connection to administer the remote queue manager.

Before you begin

If a cluster queue manager is not shown in the **Queue Managers** folder, the cluster queue manager is shown in the **Queue Manager Clusters** folder as being disconnected. You can connect to the remote cluster queue manager using the cluster information source as an intermediate queue manager. When the remote cluster queue manager is connected to IBM MQ Explorer, you can use that connection to administer the queue manager but you must first show the queue manager in the **Queue Managers** folder.

About this task

To administer a remote cluster queue manager in IBM MQ Explorer:

Procedure

1. Ensure that the remote cluster queue manager is connected to IBM MQ Explorer. For more information, see [Connecting to a remote cluster queue manager](#).
2. Right-click the queue manager, then click **Show in Queue Managers folder**.

Results

The queue manager is added to the **Queue Managers** folder and you can now administer it like any other remote queue manager.

[“Connecting to a remote cluster queue manager” on page 153](#)

You can connect IBM MQ Explorer to a remote queue manager by using the cluster information source as an intermediary queue manager.

[“Specifying a different cluster information source for IBM MQ Explorer” on page 150](#)

You can change the full repository queue manager from which IBM MQ Explorer obtains information about which queue managers belong to a cluster.

[“Administering remote queue managers” on page 111](#)

In IBM MQ Explorer, you can enable IBM MQ queue managers on a remote computer for remote administration.

[“Queue manager clusters” on page 48](#)

A cluster is a group of two or more queue managers that are logically associated and can share information with each other. Any queue manager can send a message to any other queue manager in the same cluster without you needing to set up a specific channel definition, remote queue definition, or transmission queue, because all of this information is held in the repository, to which all queue managers in the cluster have access.

Managing security and authorities

The security provisions in IBM MQ include securing channels using Transport Layer Security (TLS) and controlling access to IBM MQ objects.

About this task

You can manage both TLS security and object authorities in IBM MQ Explorer. For more information see:

- [“Securing channels with TLS” on page 155](#)
- [“Managing object authorities with an authorization service” on page 165](#)

For more information about TLS, object authorities, and other ways of securing your IBM MQ queue manager network, see [Securing IBM MQ](#) in the IBM MQ online product documentation.

Related tasks

[“Authorizing users to configure IBM MQ on Windows and Linux \(x86 and x86-64 platforms\)” on page 197](#)
IBM MQ uses the normal user and group authorizations to protect IBM MQ applications and IBM MQ administration.

[“Refreshing the authorization service information on Multiplatforms” on page 198](#)

On Multiplatforms, if you make a change to an entity, you must refresh the entity information in the authorization service. You must do this for each queue manager that is affected by the changes that you make to the entity.

[“Refreshing TLS security” on page 200](#)

You can make changes to the key repository without restarting a channel. However, the copy of the key repository that is held in memory while a channel is running will not be affected. When you refresh the cached copy of the key repository, the TLS channels that are currently running on the queue manager are updated with the new information.

[“Refreshing ESM classes \(z/OS only\)” on page 201](#)

IBM MQ for z/OS does not perform any authority checks itself; instead, it routes requests for authority checks to an external security manager (ESM).

Securing channels with TLS

The TLS (Transport Layer Security) protocol enables queue managers to communicate securely with other queue managers, or clients.

About this task

TLS Concepts

An TLS-enabled connection is secure in the following ways:

- **Authentication:** Queue managers or clients initiating an TLS-enabled connection are assured of the identity of the queue manager that they are connecting to, and queue managers that are receiving connections can check the identity of the queue manager or client that is initiating the connection.
- **Message privacy:** Using a unique session key, TLS, if configured to do so, encrypts all information exchanged over the connection. This ensures that information cannot be viewed if it is intercepted by unauthorized parties.
- **Message integrity:** The data cannot be tampered with over the connection.
- **Certificate Authority chain:** Each certificate in the Certificate Authority (CA) chain is signed by the entity that is identified by its parent certificate in the chain. At the head of the chain is the root CA certificate. The root certificate is always signed by the root CA itself. The signatures of all certificates in the chain must be verified.

Sequence overview

There are two stages to the security, as described in the following steps.

Procedure

1. When a queue manager connects to another queue manager, the two carry out a standard TLS exchange of certificates, and carry out validation checks. If the validation is successful, the connection is established. To achieve this, you must configure both of your queue managers, and the channels that they will use, with appropriate certificate settings.
2. When messages are sent from one queue manager to another queue manager along a channel, the data is generally encrypted using a session key that has been established during the certificate exchange. To achieve this you must configure the channels that you will use with appropriate CipherSpecs.

Results

Sequence Details

A typical sequence for a simple TLS connection between queue managers QM1 and QM2 is as follows:

1. QM1 connects to QM2.
2. The personal certificate that is used by QM2 is sent to QM1.
3. QM1 authenticates the personal certificate against the chain of certificate authority certificates.
4. QM1 optionally checks for certificate revocation if Online Certificate Status Protocol (OCSP) is supported on the server platform. For more information on OCSP see: [“Working with Online Certificate Status Protocol \(OCSP\)”](#) on page 42.
5. QM1 optionally checks the personal certificate against the Certificate Revocation List (CRL). For more information see: [“Configuring TLS on queue managers”](#) on page 158.
6. QM1 optionally applies a filter to only accept personal certificates that meet any defined peer names. For more information see: [“Configuring TLS channels”](#) on page 161.
7. QM1 (if all is well) accepts the personal certificate from QM2.
8. The secure connection is now established.

For more security, QM2 can request a certificate from QM1, and in that case the following steps also take place:

1. QM1 sends its assigned personal certificate to QM2.
2. QM2 applies the same checks (Steps 3, 4, and 5) as previously shown.
3. QM2, if all is well, accepts the personal certificate from QM1.

The secure connection is now established.

For more information, see [Securing IBM MQ](#) in the IBM MQ online product documentation.

Related tasks

[“Configuring TLS security for IBM MQ”](#) on page 157

To configure TLS security, you set up TLS on each queue manager and each client that uses TLS-enabled connections.

[“Configuring TLS on queue managers”](#) on page 158

After starting the IBM **strmqikm** (iKeyman) GUI, you can use it to manage TLS certificates.

Related reference

[“Authentication information properties”](#) on page 494

You can set attributes for all types of authentication information objects. Some of the attributes do not apply to all types of authentication information objects, and some attributes are specific to z/OS authentication information objects.

Configuring TLS security for IBM MQ

To configure TLS security, you set up TLS on each queue manager and each client that uses TLS-enabled connections.

About this task

For an introduction, and details on how certificates are used to establish TLS connections, see [Using SSL security with IBM MQ](#).

Setting up TLS on a queue manager

To set up TLS on a queue manager, for each queue manager that uses TLS-enabled connections:

Procedure

1. Manage the digital certificates that are used by the queue manager. For more information, see [Managing SSL certificates](#).
2. Configure the queue manager for TLS-enabled messaging. For more information, see [Configuring SSL on queue managers](#).
3. Configure channels to support secure messaging using TLS. For more information, see [Configuring SSL channels](#).

Results

Setting up TLS on an IBM MQ MQI client

To set up TLS on an IBM MQ client, for each client that uses TLS-enabled connections:

1. Manage the digital certificates that are used by the client. For more information, see [Managing SSL certificates](#).
2. Configure the client for TLS-enabled messaging. For more information, see [Configuring SSL on IBM MQ clients](#).
3. Configure the client channel definition to support secure messaging using TLS. For more information, see [Configuring SSL on IBM MQ clients](#).

For more information, see [Securing IBM MQ](#) in the IBM MQ online product documentation.

Managing TLS certificates

To manage the TLS certificates on your local computer using a GUI, use the IBM **strmqikm** command (iKeyman).

About this task

The information in this task applies to managing TLS certificates on a local computer.

Note that you cannot manage TLS certificates on remote computers using **strmqikm**.

To work with a personal certificate in **strmqikm**, you must complete the following steps.

Procedure

1. Create a key database file in the location that is specified in the queue manager's **Key repository** attribute.
2. Request and obtain from a Certificate Authority (CA) a personal certificate with the correct label and its full chain of CA certificates back to the Root certificate.

3. Add all the certificates, in the correct order, to the key repository of the queue manager using **strmqikm**.

Results

For instructions on how to use **strmqikm**, and for more information about security, see [Securing IBM MQ](#) in the IBM MQ online product documentation.

Related tasks

[“Invoking the IBM strmqikm \(iKeyman\) GUI” on page 158](#)

To manage your TLS certificates using the IBM the **strmqikm** (iKeyman) GUI, you must first open **strmqikm** from IBM MQ Explorer.

[“Configuring TLS security for IBM MQ” on page 157](#)

To configure TLS security, you set up TLS on each queue manager and each client that uses TLS-enabled connections.

Related reference

[“Queue manager properties” on page 349](#)

You can set attributes for both local and remote queue managers.

Invoking the IBM strmqikm (iKeyman) GUI

To manage your TLS certificates using the IBM the **strmqikm** (iKeyman) GUI, you must first open **strmqikm** from IBM MQ Explorer.

About this task

strmqikm

To open **strmqikm** from IBM MQ Explorer, complete the following steps.

Procedure

1. Start IBM MQ Explorer.
2. In the Navigator view, right-click **IBM MQ**, then click **Manage SSL Certificates...**

Results

The IBM **strmqikm** GUI opens.

Note that you cannot manage TLS certificates on remote computers using **strmqikm**.

For instructions on how to use **strmqikm**, and for more information about security, see [Securing IBM MQ](#) in the IBM MQ online product documentation.

Related tasks

[“Securing channels with TLS” on page 155](#)

The TLS (Transport Layer Security) protocol enables queue managers to communicate securely with other queue managers, or clients.

Configuring TLS on queue managers

After starting the IBM **strmqikm** (iKeyman) GUI, you can use it to manage TLS certificates.

Before you begin

For more information on how to start **strmqikm**, see [Invoking the IBM strmqikm \(iKeyman\) GUI](#).

About this task

The key repository is where certificates used by the queue manager are stored. On Windows, Linux, and UNIX platforms, the key repository is known as the key database file.

The location of the key repository of a queue manager is specified in the queue manager's **Key Repository** attribute. Before you can store the queue manager certificates in the key repository, you must ensure that a key database file exists in this location. If you need to create the key database file, use **strmqikm**. For more information, see [Securing IBM MQ](#) in the IBM MQ online product documentation.

Changing the queue manager key repository

About this task

In certain circumstances you might want to change the key repository; for example, to use a single location that is shared by all queue managers on one operating system.

To change a queue manager key repository location:

Procedure

1. Change the key repository location in the queue manager properties:
 - a) Open IBM MQ Explorer and expand the **Queue Managers** folder.
 - b) Right-click the queue manager, then click **Properties**.
 - c) On the **SSL** property page, edit the path in the **Key repository** field to point to your chosen directory.
 - d) On the warning dialog, click **Yes**.
2. Transfer the queue manager personal certificates to the new location using **strmqikm**.
For more information, see [Securing IBM MQ](#) in the IBM MQ online product documentation.

Authenticating certificates using Certificate Revocation Lists

About this task

Certification Authorities (CAs) can revoke certificates that are no longer trusted by publishing them in a Certification Revocation List (CRL). When a certificate is received by a queue manager or an IBM MQ MQI client, it can be checked against the CRL to ensure that it has not been revoked. CRL checking is not mandatory for TLS-enabled messaging to be achieved, but is recommended to ensure the trustworthiness of user certificates.

For more information about how to set up a CRL in this way, see [Securing IBM MQ](#) in the IBM MQ online product documentation.

To set up a connection to an LDAP CRL server:

Procedure

1. In IBM MQ Explorer, expand the queue manager.
2. Create an authentication information object of type **CRL LDAP**. For more information, see [“Creating and configuring queue managers and objects”](#) on page 27.
3. Repeat Step 2 to create as many CRL LDAP authentication information objects as you need.
4. Create a namelist and add the names of the authentication information objects that you created in Steps 2 and 3 to the new namelist. For more information, see [“Creating and configuring queue managers and objects”](#) on page 27.
5. Right-click the queue manager, then click **Properties**.

6. On the **SSL** page, in the **Revocation namelist** field, type the name of the namelist that you created in Step 4.
7. Click **OK**.



Results



The certificates that the queue manager receives can now be authenticated against the CRL held on the LDAP server.

You can add to the namelist up to 10 connections to alternative LDAP servers to ensure continuity of service if one or more LDAP servers are inaccessible.

Authenticating certificates using OCSP authentication

About this task

  On UNIX and Windows, IBM MQ TLS support checks for revoked certificates using OCSP (Online Certificate Status Protocol) or using CRLs and ARLs on LDAP (Lightweight Directory Access Protocol) servers. OCSP is the preferred method. IBM MQ classes for Java and IBM MQ classes for JMS cannot use the OCSP information in a client channel definition table file. However, you can configure OCSP as described in [Revoked certificates and OCSP](#) in the IBM MQ online product documentation.

  IBM i and z/OS do not support OCSP checking, but they do allow the generation of client channel definition tables (CCDTs) containing OCSP information.

For more information about CCDTs and OCSP, see [Client channel definition table](#) in the IBM MQ online product documentation.

To set up a connection to an OCSP server:

Procedure

1. In IBM MQ Explorer, expand the queue manager.
2. Create an authentication information object of type **OCSP**. For more information, see: [“Creating and configuring queue managers and objects”](#) on page 27.
3. Repeat Step 2 to create as many OCSP authentication information objects as you need.
4. Create a namelist and add the names of the OCSP authentication information objects that you created in Steps 2 and 3 to the new namelist. For more information, see: [“Creating and configuring queue managers and objects”](#) on page 27.
5. Right-click the queue manager, then click **Properties**.
6. On the **SSL** page, in the **Revocation namelist** field, type the name of the namelist that you created in Step 4.
7. Click **OK**.

Results

The certificates that the queue manager receives are authenticated against the OCSP responder.

The queue manager writes OCSP information to the CCDT.

Only one OCSP object can be added to the namelist because the socket library can only use one OCSP responder URL at a time.

Configuring cryptographic hardware

About this task

IBM MQ can support cryptographic hardware, and the queue manager must be configured accordingly.

To configure the queue manager for cryptographic hardware:

Procedure

1. Start IBM MQ Explorer.
2. In the Navigator view, right-click the queue manager, then click **Properties**. The Properties dialog opens.
3. On the **SSL** page, click **Configure**. The Cryptographic Hardware Settings dialog opens.
4. In the **Cryptographic Hardware Settings** dialog: All supported cryptographic cards now use PKCS #11, so ignore references to the Rainbow Cryptoswift or nCipher nFast cards. Enter the path to the PKCS #11 driver, and the token label, the token password, and the symmetric cipher setting.
5. Click **OK**.

Results

The queue manager is now configured to use the cryptographic hardware.

You can also work with certificates that are stored on PKCS #11 hardware using iKeyman.

For more information, see [Securing IBM MQ](#) in the IBM MQ online product documentation.

Related tasks

[“Configuring TLS security for IBM MQ” on page 157](#)

To configure TLS security, you set up TLS on each queue manager and each client that uses TLS-enabled connections.

[“Configuring TLS on IBM MQ MQI clients” on page 163](#)

To work with TLS on an IBM MQ client, you must use various commands.

Related reference

[“Authentication information properties” on page 494](#)

You can set attributes for all types of authentication information objects. Some of the attributes do not apply to all types of authentication information objects, and some attributes are specific to z/OS authentication information objects.

Configuring TLS channels

To configure TLS channels, you use the **SSL** page of the **Channel properties** dialog to define the cipher specification to be used. You can optionally configure a channel to accept only certificates with attributes in the distinguished name of the owner that match given values. You can also optionally configure a queue manager channel so that the queue manager refuses the connection if the initiating party does not send its own personal certificate.

About this task

To configure channels in IBM MQ Explorer, complete the following steps.

Procedure

1. Open IBM MQ Explorer.
2. In the **Navigator** view, expand the **Queue Managers** folder, then click the **Channels** folder.
3. In the **Content** view, right-click the channel, then click **Properties**.
4. In the **Properties** dialog, open the **SSL** page.

Results

Use the **SSL** page of the **Channel properties** dialog for the following tasks.

Setting message security

TLS-enabled messaging offers two methods of ensuring message security:

- Encryption ensures that if the message is intercepted, it is unreadable.
- Hash functions ensure that if the message is altered, this is detected.

The combination of these methods is called the cipher specification, or CipherSpec. The same CipherSpec must be set for both ends of a channel, otherwise TLS-enabled messaging fails. For more information, see [Securing IBM MQ](#) in the IBM MQ online product documentation.

On the **SSL** page of the **Properties** dialog, do one of the following:

- From the **Standard cipher** field, select a standard cipher.
- If you are an advanced user and you are administering a queue manager on a z/OS or IBM i platform that includes new CipherSpecs that are not the IBM MQ predefined list, enter a platform-specific value for a CipherSpec in the **Custom ciphers** field.

Filtering certificates on their owner's name

Certificates contain the distinguished name of the owner of the certificate. You can optionally configure the channel to accept only certificates with attributes in the distinguished name of the owner that match given values. To do this, select the **Accept only certificates with Distinguished Names matching these values** check box.

The attribute names that IBM MQ can filter are listed in the following table:

Attribute names	Meaning
SERIALNUMBER	Certificate serial number
MAIL	Email address
E	Email address (Deprecated in preference to MAIL)
UID or USERID	User identifier
CN	Common Name
T	Title
OU	Organizational Unit name
DC	Domain component
O	Organization name
STREET	Street / First line of address
L	Locality name
ST (or SP or S)	State or Province name
PC	Postal code / zip code
C	Country
UNSTRUCTUREDNAME	Host name
UNSTRUCTUREDADDRESS	IP address
DNQ	Distinguished name qualifier

In the **Accept only certificates with Distinguished Names matching these values** field, you can use the wildcard character (*) at the beginning or the end of the attribute value in place of any number of characters. For example, to accept only certificates from any person with a name ending with Smith working for IBM in GB, type:

```
CN=*Smith, O=IBM, C=GB
```

Authenticating parties initiating connections to a queue manager

When another party initiates a TLS-enabled connection to a queue manager, the queue manager must send its personal certificate to the initiating party as proof of identity. You can also optionally configure the queue manager channel so that the queue manager refuses the connection if the initiating party does not send its own personal certificate. To do this, on the **SSL** page of the **Channel properties** dialog, select **Required** from the **Authentication of parties initiating connections** list.

Related tasks

[“Configuring TLS security for IBM MQ” on page 157](#)

To configure TLS security, you set up TLS on each queue manager and each client that uses TLS-enabled connections.

Configuring TLS on IBM MQ MQI clients

To work with TLS on an IBM MQ client, you must use various commands.

About this task

This task introduces the commands that you use to work with TLS on an IBM MQ client. For more information, see [Securing IBM MQ](#) in the IBM MQ online product documentation.

Managing the IBM MQ client's certificates

Use the IBM **strmqikm** (iKeyman) GUI to manage your TLS certificates. For more information, see [“Invoking the IBM strmqikm \(iKeyman\) GUI” on page 158](#).

In the **strmqikm** GUI, ensure that the client key repository contains all the Certificate Authority (CA) certificates that might be required to validate certificates that are received from other queue managers.

To find out the location of the client's key repository, type the following command to examine the MQSSLKEYR environment variable:

```
echo %MQSSLKEYR%
```

Also check your application because the key repository can be set on an MQCONN call. If both values are set, the value set on the MQCONN call overrides the value of MQSSLKEYR.

Configuring the channels to use TLS

The TLS channels must be set up as described here in [“Configuring TLS channels” on page 161](#).

For more information about setting up IBM MQ client security, see [Setting up IBM MQ MQI client security](#) in the IBM MQ online product documentation.

Authenticating certificates using Certificate Revocation Lists

About this task

You can set up an IBM MQ MQI client to check certificates against CRLs on LDAP servers:

Procedure

1. On the IBM MQ server, in IBM MQ Explorer, expand the queue manager.
2. Create a new authentication information object of type **CRL LDAP**. For more information, see [“Creating and configuring queue managers and objects” on page 27](#).
3. Repeat Step 2 to create as many authentication information objects as you need.
4. Create a new namelist and add to the namelist the names of the authentication information objects that you created in Steps 2 and 3. For more information, see [“Creating and configuring queue managers and objects” on page 27](#).
5. Right-click the queue manager, then click **Properties**.
6. On the **SSL** page, in the **CRL Namelist** field, type the name of the namelist that you created in Step 4.
7. Click **OK**. All the LDAP CRL information is now written to the client channel definition table.

8. Make the client channel definition table available to the client, or, if you are using Windows Active Directory, write out the information from the client channel definition table to the Active Directory. See the [setmqscp](#) command in the IBM MQ online product documentation).

Results

For more information, see [Overview of IBM MQ MQI clients](#) in the IBM MQ online product documentation.

You can add to the namelist up to 10 connections to alternative LDAP servers to ensure continuity of service if one or more LDAP servers are inaccessible. For more information, see [Securing IBM MQ](#) in the IBM MQ online product documentation.

Authenticating certificates using OCSP authentication

About this task

You can set up an IBM MQ MQI client to check certificates against an OCSP responder. Some client environments do not support OCSP revocation checking, but all server platforms support the ability to define OCSP configuration which will be written into the client channel definition table file.

Procedure

1. On the IBM MQ server, in IBM MQ Explorer, expand the queue manager.
2. Create a new authentication information object of type **OCSP**. For more information, see [“Creating and configuring queue managers and objects”](#) on page 27.
3. Repeat Step 2 to create as many OCSP authentication information objects as you need.
4. Create a new namelist and add to the namelist the names of the OCSP authentication information objects that you created in Steps 2 and 3. For more information, see [“Creating and configuring queue managers and objects”](#) on page 27.
5. Right-click the queue manager, then click **Properties**.
6. On the **SSL** page, in the **Revocation namelist** field, type the name of the namelist that you created in Step 4.
7. Click **OK**.
8. Make the client channel definition table available to the client.

Results

For more information, see [Overview of IBM MQ MQI clients](#) in the IBM MQ online product documentation.

Only one OCSP object can be added to the namelist because the socket library can only use one OCSP responder URL at a time. For more information, see [Securing IBM MQ](#) in the IBM MQ online product documentation.

Related tasks

[“Configuring TLS security for IBM MQ”](#) on page 157

To configure TLS security, you set up TLS on each queue manager and each client that uses TLS-enabled connections.

[“Configuring TLS on queue managers”](#) on page 158

After starting the IBM **strmqikm** (iKeyman) GUI, you can use it to manage TLS certificates.

Related reference

[“Authentication information properties”](#) on page 494

You can set attributes for all types of authentication information objects. Some of the attributes do not apply to all types of authentication information objects, and some attributes are specific to z/OS authentication information objects.

Managing object authorities with an authorization service

The authorization service is an installable service that enables you to view and manage the access authorities of groups and users on IBM MQ objects. You can manage these authorities using IBM MQ Explorer.

About this task

The authorization service component that is supplied with IBM MQ is the Object Authority Manager (OAM) but you can use IBM MQ Explorer to manage authorities through other installable authorization services if you prefer.

The authorization service maintains an access control list (ACL) for each IBM MQ object to which it is controlling access. An ACL contains a list of all the group IDs that can perform operations on the object; on Windows, the ACL can contain user IDs as well as group IDs. In the authorization service, you can grant and revoke authorities for users to access queue managers and objects.

For more information about managing object authorities with the OAM, see [Object authority manager \(OAM\)](#) and [Securing IBM MQ](#) in the IBM MQ online product documentation.

For more information about granting authorities on queue managers and objects, see the following topics:

- [Granting the Create authority](#)
- [Granting authorities on a queue manager](#)
- [Granting authorities on a specific object](#)
- [Granting authorities on multiple objects](#)

Related concepts

[“Authorities you can set on IBM MQ objects” on page 182](#)

You can set authorities for users and groups accessing different IBM MQ objects.

[“Authority records” on page 179](#)

An authority record is the set of authorities that have been granted to a particular user or group of users (entities) on a named object.

[“Accumulated authorities” on page 178](#)

Accumulated authorities are the total authorities that a user or group has to perform an operation on an object.

[“Users and groups \(entities\) in the authorization service” on page 179](#)

In the authorization service, authorities are granted to users (also known as principals when the user name is fully qualified with the domain name) or groups of users for accessing IBM MQ objects. Users and groups are collectively known as entities in the authorization service. You grant a set of authorities to an entity by creating an authority record.

Related tasks

[“Enabling installed plug-ins” on page 255](#)

If a new plug-in that you install in IBM MQ Explorer is not enabled by default, you can enable it by using the Preferences dialog.

Granting the Create authority

To create a new object on a queue manager, the user who performs the operation must have authority to create that type of object on the queue manager.

About this task

The authority can be granted to the group to which the user belongs (in which case, all members of the group are granted the Create authority) or, on Windows queue managers only, to an individual user.

The user can have the authority to create any type of object on the queue manager or just to create specific types of objects; for example, channels, queues, and listeners only.

Note that the ability to create a queue indirectly grants full administrative rights. Do not grant Create authority to ordinary users or applications.

To grant the authority to a group or user to create objects on a queue manager, complete the following steps.

Procedure

1. In the Navigator view, right-click the queue manager, then click **Object Authorities > Manage Create Authorities...** The Manage Create Authorities dialog opens.
2. Windows queue managers only: if you are granting the authority to an individual user, click the **Users** tab.
3. Click **New...** The Add Authorities dialog opens.
4. Enter the name of the group or user, as appropriate.
5. Select the check boxes for the objects for which you want to grant the Create authority, then click **OK**.

Results

An authority record for the group or user is added to the table and the Create authorities that you granted are shown.

If the group or user already has Create authorities for some of the objects on the queue manager, select the existing authority record and edit it. If you add a new authority record for a user or group that already has an authority record on the object, you are prompted to confirm that you want to overwrite the existing authority record.

Related concepts

[“Users and groups \(entities\) in the authorization service” on page 179](#)

In the authorization service, authorities are granted to users (also known as principals when the user name is fully qualified with the domain name) or groups of users for accessing IBM MQ objects. Users and groups are collectively known as entities in the authorization service. You grant a set of authorities to an entity by creating an authority record.

[“Authorities you can set on IBM MQ objects” on page 182](#)

You can set authorities for users and groups accessing different IBM MQ objects.

Related tasks

[“Granting authorities on a queue manager” on page 168](#)

To perform an operation on a queue manager, the user must have authority to perform that particular operation on the queue manager.

[“Granting authorities on a specific object” on page 169](#)

A user must have the correct authorities to perform operations on objects; for example, to browse the messages on a queue.

[“Granting authorities on multiple objects” on page 170](#)

A user must have the correct authorities to perform operations on objects; for example, to browse the messages on a queue. You can grant the same set of authorities to multiple objects on a queue manager by using generic profiles.

Granting role-based authorities on a queue manager

A user must have the correct authorities to perform operations on objects; you can assign these authorities individually, but if a user needs either read-only access or full administrative access to all the objects hosted by a queue manager, this can be granted in a single action.

About this task

Note: This procedure grants the requested access in addition to whatever access the user or group currently has. If you grant read-only access to a user or group, that user or group does not lose any existing administrative authorities.

To grant a group or user either read-only access or full administrative access to all the objects hosted by a queue manager, complete the following steps.

Procedure

1. In the Navigator view, right-click the queue manager, then click **Object Authorities > Add Role Based Authorities...** The Add Role Based Authorities dialog opens.
2. Windows queue managers only: if you are granting the authority to an individual user, click **User** and enter the user name.
3. If you are granting the authority to a group, click **Group** and enter the group name.
4. Select the appropriate radio button to grant read only access or full administrative access.
5. If you want to allow the user or group to browse messages on the queues hosted by the queue manager, select the **Permit reading of messages on queues** check box.
6. Equivalent commands to grant the requested authorities are displayed in the **Command preview** pane. You can copy one or more commands and paste them into a script or onto the command line.
7. Click **OK**.

Results

The requested authorities are granted to the user or group.

Note: On IBM i, you might also need to change access authorities to allow the user to issue the commands you have generated. Do this using the **GRTOBJAUT** command.

Related concepts

[“Users and groups \(entities\) in the authorization service” on page 179](#)

In the authorization service, authorities are granted to users (also known as principals when the user name is fully qualified with the domain name) or groups of users for accessing IBM MQ objects. Users and groups are collectively known as entities in the authorization service. You grant a set of authorities to an entity by creating an authority record.

Related tasks

[“Granting authorities on a specific object” on page 169](#)

A user must have the correct authorities to perform operations on objects; for example, to browse the messages on a queue.

[“Granting authorities on multiple objects” on page 170](#)

A user must have the correct authorities to perform operations on objects; for example, to browse the messages on a queue. You can grant the same set of authorities to multiple objects on a queue manager by using generic profiles.

[“Granting the Create authority” on page 166](#)

To create a new object on a queue manager, the user who performs the operation must have authority to create that type of object on the queue manager.

[“Granting the authority to connect to a queue manager” on page 171](#)

Before a user can access a queue manager's objects, the user must connect to the queue manager. The user must, therefore, have authority to connect to that queue manager.

Granting authorities on a queue manager

To perform an operation on a queue manager, the user must have authority to perform that particular operation on the queue manager.

About this task

The user can have the authority to perform any operation on the queue manager or just to perform specific operations; for example, to connect to the queue manager, to delete the queue manager, or to display the queue manager's attributes.

To grant authorities to a group or user to perform operations on a queue manager, complete the following steps.

Procedure

1. In the Navigator view, right-click the queue manager, then click **Object Authorities > Manage Queue Manager Authority Records...** The Manage Authority Records dialog opens.
2. Windows queue managers only: if you are granting the authority to an individual user, click the **Users** tab.
3. Click **New...** The Add Authorities dialog opens.
4. Enter the name of the group or user, as appropriate.
5. Select the check boxes for the authorities that you want to grant, then click **OK**.

Results

An authority record for the group or user is added to the table and the authorities that you granted are shown.

If the user or group already has some authorities on the queue manager, select the existing authority record and edit it. If you add a new authority record for a user or group that already has an authority record on the object, you are prompted to confirm that you want to overwrite the existing authority record.

Related concepts

[“Users and groups \(entities\) in the authorization service” on page 179](#)

In the authorization service, authorities are granted to users (also known as principals when the user name is fully qualified with the domain name) or groups of users for accessing IBM MQ objects. Users and groups are collectively known as entities in the authorization service. You grant a set of authorities to an entity by creating an authority record.

[“Authorities you can set on IBM MQ objects” on page 182](#)

You can set authorities for users and groups accessing different IBM MQ objects.

Related tasks

[“Granting authorities on a specific object” on page 169](#)

A user must have the correct authorities to perform operations on objects; for example, to browse the messages on a queue.

[“Granting authorities on multiple objects” on page 170](#)

A user must have the correct authorities to perform operations on objects; for example, to browse the messages on a queue. You can grant the same set of authorities to multiple objects on a queue manager by using generic profiles.

[“Granting the Create authority” on page 166](#)

To create a new object on a queue manager, the user who performs the operation must have authority to create that type of object on the queue manager.

Granting authorities on a specific object

A user must have the correct authorities to perform operations on objects; for example, to browse the messages on a queue.

About this task

To grant a user or group of users authority to perform operations on a specific object, complete the following steps.

Procedure

1. In the Content view, right-click the object, then click **Object Authorities > Manage Authority Records**. The Manage Authority Records dialog opens.
2. Expand the **Specific Profiles** folder. Only one profile is displayed because only one specific profile can match one object. If you open the Manage Authority Records dialog from a folder in the Navigator view, a specific profile for each of the objects in the folder is displayed in the **Specific Profiles** folder.
3. Click the profile that is displayed in the **Specific Profiles** folder. The authority records that have been granted on the object are displayed.
4. Windows queue managers only: if you are granting the authority to an individual user, click the **Users** tab.
5. Click **New...** The Add Authorities dialog opens.
6. Enter the name of the group or user, as appropriate.
7. Select the check boxes for the authorities that you want to grant on the object, then click **OK**.

Results

An authority record for the user or group is added to the table and the authorities that you granted are shown in the authority record.

If the user or group already has some authorities for the object, select the existing authority record and edit it. If you add a new authority record for a user or group that already has an authority record on the object, you are prompted to confirm that you want to overwrite the existing authority record.

Related concepts

[“Generic and specific profiles” on page 180](#)

When you manage authorities for a folder of objects (for example, the Queues folder) using the Manage Authority Records dialog, you grant authorities against profiles instead of granting authorities on specific objects.

[“Users and groups \(entities\) in the authorization service” on page 179](#)

In the authorization service, authorities are granted to users (also known as principals when the user name is fully qualified with the domain name) or groups of users for accessing IBM MQ objects. Users and groups are collectively known as entities in the authorization service. You grant a set of authorities to an entity by creating an authority record.

[“Authorities you can set on IBM MQ objects” on page 182](#)

You can set authorities for users and groups accessing different IBM MQ objects.

Related tasks

[“Granting authorities on multiple objects” on page 170](#)

A user must have the correct authorities to perform operations on objects; for example, to browse the messages on a queue. You can grant the same set of authorities to multiple objects on a queue manager by using generic profiles.

[“Granting the Create authority” on page 166](#)

To create a new object on a queue manager, the user who performs the operation must have authority to create that type of object on the queue manager.

Granting authorities on multiple objects

A user must have the correct authorities to perform operations on objects; for example, to browse the messages on a queue. You can grant the same set of authorities to multiple objects on a queue manager by using generic profiles.

About this task

To grant a user or group the same set of authorities on multiple objects, complete the following steps.

Procedure

1. In the Navigator view, on the queue manager that hosts the objects, right-click the folder that contains the objects, then click **Object Authorities > Manage Authority Records....** The Manage Authority Records dialog opens.
2. You can use an existing generic profile or create a new generic profile:
 - If there is an existing generic profile that matches the objects, expand the **Generic Profiles** folder, click the generic profile, then click **New > User Authority...** or **New > Group Authority...** The Add Authorities dialog opens.
 - If there is no existing generic profile that matches the objects, right-click the **Generic Profiles** folder, then click **New > User Authority Using New Profile...** or **New > Group Authority Using New Profile....** The Add Using Generic Profile dialog opens.
3. Enter the name of the user or group..
4. Type a name for the profile using wildcard characters. The name of the profile must match the names of all of the objects to which you want the profile to apply.
5. Select the check boxes for the authorities that you want to grant on the objects, then click **OK**.

Results

An authority record for the user or group is added to the table and the authorities that you granted are shown.

If the user or group already has some authorities for the object, select the existing authority record and edit it. If you add a new authority record for a user or group that already has an authority record on the object, you are prompted to confirm that you want to overwrite the existing authority record.

Related concepts

[“Generic and specific profiles” on page 180](#)

When you manage authorities for a folder of objects (for example, the Queues folder) using the Manage Authority Records dialog, you grant authorities against profiles instead of granting authorities on specific objects.

[“Users and groups \(entities\) in the authorization service” on page 179](#)

In the authorization service, authorities are granted to users (also known as principals when the user name is fully qualified with the domain name) or groups of users for accessing IBM MQ objects. Users and groups are collectively known as entities in the authorization service. You grant a set of authorities to an entity by creating an authority record.

[“Authorities you can set on IBM MQ objects” on page 182](#)

You can set authorities for users and groups accessing different IBM MQ objects.

Related tasks

[“Granting authorities on a specific object” on page 169](#)

A user must have the correct authorities to perform operations on objects; for example, to browse the messages on a queue.

Related reference

[“Wildcards used in generic profiles” on page 188](#)

You can use some wildcard characters in generic profiles.

Granting the authority to connect to a queue manager

Before a user can access a queue manager's objects, the user must connect to the queue manager. The user must, therefore, have authority to connect to that queue manager.

About this task

Any authorities granted to the user on the queue manager's objects are irrelevant unless the user can connect to the queue manager.

When you view the authority records of objects on a queue manager for which the user does not have Connect authority, the Find Accumulated Authorities dialog displays a message to warn you that the authorities will have no effect until you grant Connect authority to the user or to a group to which the user belongs.

To grant Connect authority for a queue manager to a user or group, complete the following steps.

Procedure

1. In the Navigator view, right-click the queue manager, then click **Object Authorities > Manage Queue Manager Authority Records...** The Manage Authority Records dialog opens.
2. Highlight the record for the user or group to which you want to add the Connect authority, then click **Edit...** The Edit Authorities dialog opens.
3. Select the **Connect** check box, then click **OK**.

Results

The user now has Connect access to the queue manager. When the user accesses the queue manager's objects, the authorities that you have granted to the user take effect.

Related concepts

[“Authorities you can set on IBM MQ objects” on page 182](#)

You can set authorities for users and groups accessing different IBM MQ objects.

Related tasks

[“Granting authorities on a queue manager” on page 168](#)

To perform an operation on a queue manager, the user must have authority to perform that particular operation on the queue manager.

[“Granting authorities on a specific object” on page 169](#)

A user must have the correct authorities to perform operations on objects; for example, to browse the messages on a queue.

[“Granting authorities on multiple objects” on page 170](#)

A user must have the correct authorities to perform operations on objects; for example, to browse the messages on a queue. You can grant the same set of authorities to multiple objects on a queue manager by using generic profiles.

[“Granting the Create authority” on page 166](#)

To create a new object on a queue manager, the user who performs the operation must have authority to create that type of object on the queue manager.

Comparing the authorities of two entities

In the authorization service, you can compare the authorities that have been granted to two groups of users.

About this task

An example of the authorities that you could compare is to compare the authorities of group AppDev6 with the authorities of SysDev6 on queue Q_STOCKS_5.

On Windows queue managers, you can also compare the authorities that have been granted to two individual users, or compare the authorities of a group with the authorities of an individual user.

To compare the authorities of two groups or users, complete the following steps.

Procedure

1. In the Content view, right-click the object on which the two groups or users have authorities, then click **Object Authorities > Manage Authority Records...** The Manage Authority Records dialog opens.
2. Click the profile (generic profile or specific profile) that matches the objects on which the two groups or users have authorities. The authority records associated with the profile are displayed.
3. Click the authority record of one of the groups or users, then click **Compare** The Compare Authority Records dialog opens.
4. Enter the name of the group or user with which you want to compare authorities, then click **Compare**. The two groups or users and their authorities are displayed in the table.
5. Optional: To show only the authorities that are set differently, select the **Show differences only** check box. The authorities that are the same for both groups or users are hidden so that you can see the differences more easily. In the following figure, the Compare Authority Records dialog shows that the only differences between the authority records of the user called User500 and the group called AppDev6 are that the Browse, Get, Inquire, and Set authorities have been granted explicitly to AppDev6 but not to User500.

Name	Browse	Get	Inquire	Set
User500@HAPPYLAPPY				
AppDev6@HAPPYLAPPY	✓	✓	✓	✓

Results

The dialog displays only the authority records for the entities on the object. The dialog does not display authorities that the user or group might inherit from other sources (the accumulated authorities). For more information about comparing accumulated authorities, see [Comparing the accumulated authorities of two entities](#).

Related concepts

[“Users and groups \(entities\) in the authorization service” on page 179](#)

In the authorization service, authorities are granted to users (also known as principals when the user name is fully qualified with the domain name) or groups of users for accessing IBM MQ objects. Users and groups are collectively known as entities in the authorization service. You grant a set of authorities to an entity by creating an authority record.

Related tasks

[“Granting authorities on a specific object” on page 169](#)

A user must have the correct authorities to perform operations on objects; for example, to browse the messages on a queue.

Comparing the accumulated authorities of two entities

You can compare the accumulated authorities on an object of two users, two groups, or a user with a group.

About this task

To compare the accumulated authorities of two entities, complete the following steps.

Procedure

1. Display the accumulated authorities for a user or group on an object. For more information, see [Finding the accumulated authorities of an entity on an object](#).
2. Click the accumulated authorities row of the table to highlight it, then click **Compare**. The Compare Accumulated Authorities dialog opens.
3. Enter the name and type of the entity with which you want to compare the accumulated authorities, then click **Compare**. The two sets of accumulated authorities are displayed in the table.
4. Optional: Select the **Show accumulated differences only** check box to show only the authorities that are different. For example, in the following figure, the Compare Accumulated Authority Records dialog shows that in the comparison between the user called User500 and the group called mqm, the only difference is that mqm has the Put authority but User500 does not.

Compare Group:

With the: ☒ Group ☐ User

Called:

☒ Show accumulated differences only

☐ Show authority records

Name	Put
+ User500@HAPPYLAPPY	
+ mqm	✓

5. Optional: Select the **Show authorities records** check box to expand both sets of accumulated authorities row to display the authority records that contribute to the accumulated authorities. The following figure shows the comparison between the user called User500 and the group called mqm with the authority records displayed.

Compare Group:

With the: ☒ Group ☐ User

Called:

☒ Show accumulated differences only

☒ Show authority records

Name	Put
- User500@HAPPYLAPPY	
User500@HAPPYLAPPY	
AppDev6@HAPPYLAPPY	
- mqm	✓
mqm@HAPPYLAPPY	✓

Results

The dialog displays the accumulated authorities and the authority records that contribute to the accumulated authorities. You cannot edit the authority records from this dialog.

Related concepts

[“Accumulated authorities” on page 178](#)

Accumulated authorities are the total authorities that a user or group has to perform an operation on an object.

[“Users and groups \(entities\) in the authorization service” on page 179](#)

In the authorization service, authorities are granted to users (also known as principals when the user name is fully qualified with the domain name) or groups of users for accessing IBM MQ objects. Users and groups are collectively known as entities in the authorization service. You grant a set of authorities to an entity by creating an authority record.

Related tasks

[“Comparing the authorities of two entities” on page 172](#)

In the authorization service, you can compare the authorities that have been granted to two groups of users.

Finding the authorities of a user or group on an object

You can search the authorization service for authority records or accumulated authorities that have been granted to groups or users (entities) on a queue manager's objects. If the group or user does not have an authority record on the specified objects, no results are displayed.

About this task

To find the authorities, complete the following steps.

Procedure

1. In the Navigator view, right-click the queue manager, then click **Object Authorities > Find Authorities**. The Find Authorities dialog opens.
2. Select the type of information that you want to display:
 - To view the authorities that have been explicitly granted to the group or user, click **Authority records**.
 - To view the authorities that have accumulated for the group or user, click **Accumulated authorities**.
3. In the **Entity type** field, select the entity for whom you are finding the authorities:
 - To view the authorities for a specific user, click **A user**. If **Authority records** is selected, this option is available on Windows queue managers only.
 - To view the authorities for a specific group of users, click **A group**.
 - To view the authorities for a group or a user of a particular name, click **A user or a group**. This option is available on Windows queue managers only.
 - To view the authorities for all users, click **All users**. This option is available on Windows queue managers only.
 - To view the authorities for all groups, click **All groups**.
 - To view the authorities for all entities, click **All users and groups**. This option is available on Windows queue managers only.
4. In the **Entity name** field, type the name of the entity.
5. In the **Object type** field, select the type of object on which the authorities were granted.
6. In the **Profile type** field, select the type of profile that the object's name must match:
 - To find authorities on a specific object, click **Specific profile**.
 - To find authorities on multiple objects, click **Generic profile**. The generic profile must already exist.
7. In the **Profile name** field, enter the name of the profile that the object name must match.
8. Click **Find**.

Results

The authority records or accumulated authorities are displayed in the table.

You can edit or remove the authority records that are displayed in the table. Be aware, however, that removing an authority record might revoke authorities from that user or group (or from any users in the group) that is associated with that record.

Related concepts

[“Accumulated authorities” on page 178](#)

Accumulated authorities are the total authorities that a user or group has to perform an operation on an object.

[“Generic and specific profiles” on page 180](#)

When you manage authorities for a folder of objects (for example, the Queues folder) using the Manage Authority Records dialog, you grant authorities against profiles instead of granting authorities on specific objects.

[“Users and groups \(entities\) in the authorization service” on page 179](#)

In the authorization service, authorities are granted to users (also known as principals when the user name is fully qualified with the domain name) or groups of users for accessing IBM MQ objects. Users and groups are collectively known as entities in the authorization service. You grant a set of authorities to an entity by creating an authority record.

Related tasks

[“Granting authorities on a specific object” on page 169](#)

A user must have the correct authorities to perform operations on objects; for example, to browse the messages on a queue.

[“Granting authorities on multiple objects” on page 170](#)

A user must have the correct authorities to perform operations on objects; for example, to browse the messages on a queue. You can grant the same set of authorities to multiple objects on a queue manager by using generic profiles.

[“Granting authorities on a queue manager” on page 168](#)

To perform an operation on a queue manager, the user must have authority to perform that particular operation on the queue manager.

Finding the accumulated authorities of an entity on an object

You can find and view the accumulated authorities of an entity. The accumulated effect of the authorities of an entity on an object affects whether the entity can perform operations on the object.

About this task

When you view the authority records that have been created on a specific object (for example, a queue called Q2) in the Manage Authority Records dialog, you can see what authorities have been explicitly granted to the user or group (the entity) on that object; you can also see which generic profiles apply to that object and whether the entity has authority records against any of the generic profiles. You cannot, however, easily see the accumulated effect of those authorities, which is ultimately what affects whether the entity can perform operations on the object.

You can find and view the accumulated authorities of an entity on an object in any of the following ways:

- In the Manage Authority Records dialog, click the authority record for the entity, then click **Accumulated Authorities...** For more information about opening the Manage Authority Records dialog, see [“Granting authorities on a specific object” on page 169](#) or [“Granting authorities on multiple objects” on page 170](#).
- In the Content view, right-click the object, then click **Object Authorities > Find Accumulated Authorities...**
- In the Navigator view, right-click the queue manager, then click **Object Authorities > Find Authorities...** For more information about finding accumulated authorities in the Find Authorities dialog, see [Finding the authorities of a user or group on an object](#).

The accumulated authorities for the entity are displayed in the first row of the table; the other rows display all of the authority records that contribute to the accumulated authorities. If one authority record contains an authority to perform a particular operation (for example, to put messages on the queue), the accumulated authorities allow the entity to perform that operation.

You can edit one or more of the authority records that contribute to the accumulated authorities. You can also remove an authority record from in the Find Accumulated Authorities dialog. Be aware, however, that removing an authority record might revoke authorities from the user or group (or from any users in the group) that is associated with that record.

Related concepts

[“Accumulated authorities” on page 178](#)

Accumulated authorities are the total authorities that a user or group has to perform an operation on an object.

[“Users and groups \(entities\) in the authorization service” on page 179](#)

In the authorization service, authorities are granted to users (also known as principals when the user name is fully qualified with the domain name) or groups of users for accessing IBM MQ objects. Users and groups are collectively known as entities in the authorization service. You grant a set of authorities to an entity by creating an authority record.

[“Generic and specific profiles” on page 180](#)

When you manage authorities for a folder of objects (for example, the Queues folder) using the Manage Authority Records dialog, you grant authorities against profiles instead of granting authorities on specific objects.

Related tasks

[“Granting authorities on multiple objects” on page 170](#)

A user must have the correct authorities to perform operations on objects; for example, to browse the messages on a queue. You can grant the same set of authorities to multiple objects on a queue manager by using generic profiles.

[“Granting authorities on a specific object” on page 169](#)

A user must have the correct authorities to perform operations on objects; for example, to browse the messages on a queue.

Determining why an entity has certain authorities

An entity's authorities can accumulate from several sources so it is useful to be able to find out which authority records contributed to an entity's accumulated authorities.

About this task

After determining why an entity has certain authorities, you can then change the accumulated authorities in one or more of the authority records as appropriate.

To determine why an entity has certain authorities on an object, complete the following steps.

Procedure

1. In the Content view, right-click the object, then click **Object Authorities > Find Accumulated Authorities...** The Find Accumulated Authorities dialog opens.
2. Select the type of entity and type the name of the entity. The table displays the entity's accumulated authorities and the authority records that contribute to them.
3. Look down the column of the authority (for example, the **Put** column) to determine which authority record has caused the entity to have that accumulated authority.

Results

When you have determined which authority records have contributed to the group or user's accumulated authorities, you can edit one or more of the authority records to change the accumulated authorities (be aware that changes you make could be inherited by other groups or users as well).

You can also remove an authority record from in the Find Accumulated Authorities dialog. Be aware, however, that removing an authority record might revoke authorities from the user or group (or from any users in the group) that is associated with that record.

Related concepts

[“Accumulated authorities” on page 178](#)

Accumulated authorities are the total authorities that a user or group has to perform an operation on an object.

[“Users and groups \(entities\) in the authorization service” on page 179](#)

In the authorization service, authorities are granted to users (also known as principals when the user name is fully qualified with the domain name) or groups of users for accessing IBM MQ objects. Users and groups are collectively known as entities in the authorization service. You grant a set of authorities to an entity by creating an authority record.

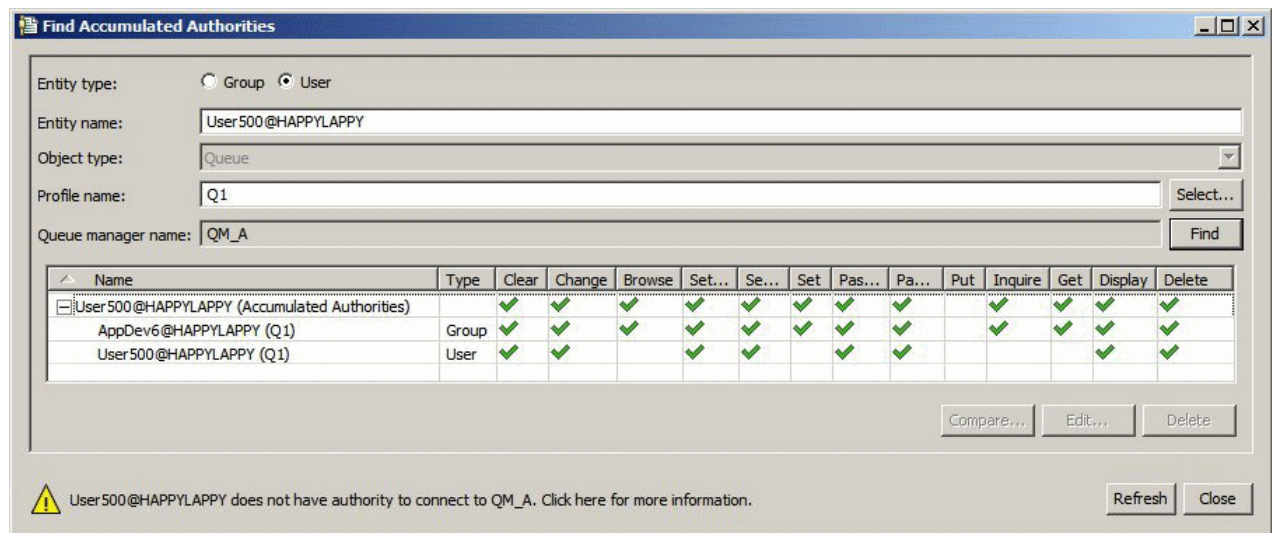
Accumulated authorities

Accumulated authorities are the total authorities that a user or group has to perform an operation on an object.

A user can be granted authorities on an object from the following sources:

- An authority record that has been created on the object for the user (Windows only).
- An authority record that has been created on the object for a group to which the user belongs.
- An authority record that has been created for the user against a generic profile that matches the object (Windows only).
- An authority record that has been created for a group to which the user belongs against a generic profile that matches the object.

If a user is granted an authority (for example, the authority to put messages on a queue called Q1) from just one of these sources, the user has that authority, even if authority records from other sources do not grant that authority. For example, the following figure shows that the user called User500, who belongs to group AppDev6, does not have authority to put messages on Q1 because the Put authority has not been granted to User500 or to AppDev6. User500, however, does have authority to get messages from Q1 because the Get authority has been granted to AppDev6 so User500 inherits the Get authority.



In the figure, the first row of the table in the Find Accumulated Authorities dialog shows the accumulated authorities of User500. The next two rows show the authority records that contribute to the accumulated authorities. In the scenario shown in the figure, the authority record for User500 does not contain the Put and Get authorities; the authority record for AppDev6, however, contains the Get authority. Therefore, the accumulated authorities for User500 show that User500 has Get authority but not Put authority on queue Q1.

The warning message in the Find Accumulated Authorities dialog shows that although User500 has some authorities to perform operations on queue Q1, User500 does not have authority to connect to the queue manager that hosts Q1.

Related concepts

[“Authority records” on page 179](#)

An authority record is the set of authorities that have been granted to a particular user or group of users (entities) on a named object.

[“Users and groups \(entities\) in the authorization service” on page 179](#)

In the authorization service, authorities are granted to users (also known as principals when the user name is fully qualified with the domain name) or groups of users for accessing IBM MQ objects. Users and groups are collectively known as entities in the authorization service. You grant a set of authorities to an entity by creating an authority record.

Related tasks

[“Granting the Create authority” on page 166](#)

To create a new object on a queue manager, the user who performs the operation must have authority to create that type of object on the queue manager.

Authority records

An authority record is the set of authorities that have been granted to a particular user or group of users (entities) on a named object.

On objects on Windows, you can create authority records for individual users and for groups of users. On UNIX, Linux, and IBM i, you can create authority records only for groups of users; if you grant authorities to an individual user, the authorization service creates or updates the authority record for the user's primary group so that the same authorities are granted to all the users in the group.

To be able to perform operations on an object or a queue manager, an entity (a user or a group) must have an authority record that contains the authorities to perform those operations. For example, for a user called User337 to be able to put messages on queue Q1, User337 or a group to which User337 belongs must have an authority record that contains the Put authority.

You can grant authorities on single objects by creating an authority record against a specific profile, or you can grant authorities on multiple objects by creating an authority record against a generic profile. Because you can create authority records for individual users and for groups, and you can create authority records against generic profiles which can apply to multiple objects, the authorities that an individual user has on a particular object can accumulate from several sources.

Related concepts

[“Accumulated authorities” on page 178](#)

Accumulated authorities are the total authorities that a user or group has to perform an operation on an object.

[“Generic and specific profiles” on page 180](#)

When you manage authorities for a folder of objects (for example, the Queues folder) using the Manage Authority Records dialog, you grant authorities against profiles instead of granting authorities on specific objects.

Related tasks

[“Determining why an entity has certain authorities” on page 177](#)

An entity's authorities can accumulate from several sources so it is useful to be able to find out which authority records contributed to an entity's accumulated authorities.

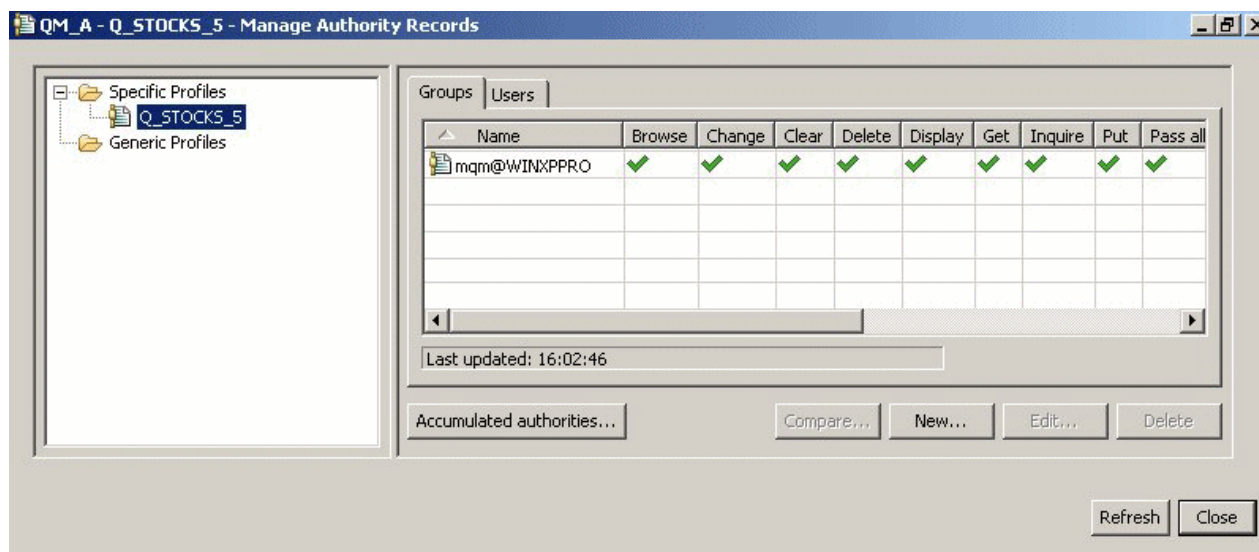
Users and groups (entities) in the authorization service

In the authorization service, authorities are granted to users (also known as principals when the user name is fully qualified with the domain name) or groups of users for accessing IBM MQ objects. Users and

groups are collectively known as entities in the authorization service. You grant a set of authorities to an entity by creating an authority record.

On objects on Windows, you can create authority records for individual users and for groups of users. On UNIX, Linux, and IBM i, you can create authority records only for groups of users; if you grant authorities to an individual user, the authorization service creates or updates the authority record for the user's primary group so that the same authorities are granted to all the users in the primary group.

The following figure shows the authority record for the mqm group on a queue called Q_STOCKS_5. Q_STOCKS_5 is a queue on a Windows queue manager so it is possible to view authority records that have been created for individual users. If the queue were hosted on a UNIX, Linux, or IBM i queue manager, there would be no **Users** tab available in the dialog.



The users and groups that are displayed in IBM MQ Explorer are defined in the operating system that hosts the queue manager and objects. You cannot, therefore, create or delete entities from within the IBM MQ Explorer itself. If you make a change to an entity while IBM MQ Explorer is running, you must refresh the authorization service to pick up the changes; for more information, see [Refreshing authorization service information](#).

Entities can be granted authorities explicitly and also by inheritance. For more information about how entities can inherit authorities, see [Accumulated authorities](#).

On Windows, delete the authority records corresponding to a particular Windows user account before deleting that user account. It is impossible to remove the authority records after removing the Windows user account.

Related concepts

[“Authority records” on page 179](#)

An authority record is the set of authorities that have been granted to a particular user or group of users (entities) on a named object.

[“Accumulated authorities” on page 178](#)

Accumulated authorities are the total authorities that a user or group has to perform an operation on an object.

Generic and specific profiles

When you manage authorities for a folder of objects (for example, the Queues folder) using the Manage Authority Records dialog, you grant authorities against profiles instead of granting authorities on specific objects.

Profiles define the name and type of object to which the authorities will apply. A specific profile exactly matches the name of the object, while a generic profile matches one or more objects using wildcard characters.

Specific profiles

A specific profile applies only to the object of that name and type. To grant or revoke an authority on a single object, you select the relevant specific profile and create or edit the authority records for that profile.

For example, to grant group AppDev6 the authority to put messages on queue Q.STOCKS.5, select the specific profile called Q . STOCKS . 5 and create or edit the authority record for group AppDev6. The authority record will apply only to the queue called Q.STOCKS.5.

Objects of type queue or topic with names that match the profile name do not have to exist when the command is issued.

Generic profiles

A generic profile is a profile that you have created to associate with more than one object of the same type. You can grant authorities to a set of objects at the same time by creating an authority record against the generic profile. For example, to grant group AppDev6 the authority to put messages on any queue with a name that starts with Q.STOCKS., grant the authority using a generic profile that is named Q . STOCKS . * For more information about wildcards, see [Wildcards used in generic profiles](#).

Objects with names that match the profile name do not have to exist when the command is issued.

Related concepts

[“Users and groups \(entities\) in the authorization service” on page 179](#)

In the authorization service, authorities are granted to users (also known as principals when the user name is fully qualified with the domain name) or groups of users for accessing IBM MQ objects. Users and groups are collectively known as entities in the authorization service. You grant a set of authorities to an entity by creating an authority record.

Related tasks

[“Granting authorities on a specific object” on page 169](#)

A user must have the correct authorities to perform operations on objects; for example, to browse the messages on a queue.

[“Granting authorities on multiple objects” on page 170](#)

A user must have the correct authorities to perform operations on objects; for example, to browse the messages on a queue. You can grant the same set of authorities to multiple objects on a queue manager by using generic profiles.

Related reference

[“Wildcards used in generic profiles” on page 188](#)

You can use some wildcard characters in generic profiles.

Authorization service control commands

IBM MQ Explorer performs the same functions as the IBM MQ control commands setmqaut, dspmqaut, and dmpmqaut.

The following table shows the authorities in IBM MQ Explorer and the equivalent parameters when you use the control commands.

Authority	Control command
Alternate user authority	altusr
Browse	browse
Change	chg
Clear	clr
Connect	connect
Create	crt

Authority	Control command
Ctrl	ctrl
Ctrlx	ctrlx
Delete	dlt
Display	dsp
Get	get
Put	put
Inquire	inq
Pass all context	passall
Pass identity context	passid
Set	set
Set all context	setall
Set identity context	setid
System	system

Related tasks

[“Granting authorities on a queue manager” on page 168](#)

To perform an operation on a queue manager, the user must have authority to perform that particular operation on the queue manager.

[“Granting authorities on a specific object” on page 169](#)

A user must have the correct authorities to perform operations on objects; for example, to browse the messages on a queue.

[“Granting authorities on multiple objects” on page 170](#)

A user must have the correct authorities to perform operations on objects; for example, to browse the messages on a queue. You can grant the same set of authorities to multiple objects on a queue manager by using generic profiles.

Authorities you can set on IBM MQ objects

You can set authorities for users and groups accessing different IBM MQ objects.

The following table lists the authorities that you can set for users and groups accessing different IBM MQ objects. Some authorities can be set against specific objects only; the table shows whether each authority is valid for each object.

Authority	Description	Queue manager	Remote queue manager	Queue	Process definitions	Name list	Authentication information	Channel	Client-connection channel	Service	Listener
Alternate user ID	Use another user's ID to open queues and put messages on queues.	Yes	No	No	No	No	No	No	No	No	No
Browse	Browse messages on a queue.	No	No	Yes	No	No	No	No	No	No	No
Change	Change the attributes of the object.	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Clear	Clear the messages from the queue.	No	No	Yes	No	No	No	No	No	No	No
Connect	Allow the application to connect to the queue manager.	Yes	No	No	No	No	No	No	No	No	No

Authority	Description	Queue manager	Remote queue manager	Queue	Process definitions	Namespace	Authentication information	Channel	Client-connection channel	Service	Listener
Create	Create objects of the specified type on the queue manager.	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Ctrl	Start, stop, and ping the channel.	No	No	No	No	No	No	Yes	No	Yes	Yes
Ctrlx	Reset or resolve the channel.	No	No	No	No	No	No	Yes	No	No	No
Delete	Delete the object.	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Display	Display the attributes or status of the object.	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Get	Get messages from the queue.	No	No	Yes	No	No	No	No	No	No	No
Put	Put messages on the queue.	No	Yes	Yes	No	No	No	No	No	No	No

Authority	Description	Queue manager	Remote queue manager	Queue	Process definitions	Namespace	Authentication information	Channel	Client-connection channel	Service	Listener
Inquire	Display the attributes or status of the object.	Yes	No	Yes	Yes	Yes	Yes	No	No	No	No
Pass all context	Allow the application to pass all the context fields from the request message to a message that the application is putting on the queue.	No	Yes	Yes	No	No	No	No	No	No	No

Author ity	Descri ption	Queue manag er	Remot e queue manag er	Queue	Proces s definit ions	Namel ist	Authe nticati on inform ation	Chann el	Client- conne ction chann el	Servic e	Listen er
Pass identit y context	Allow the application to pass the identity context fields from the request message to the message that the application is putting on a queue.	No	Yes	Yes	No	No	No	No	No	No	No
Set	Set attributes on the queue.	Yes	No	Yes	Yes	No	No	No	No	No	No
Set all context	Allow the application to set the identity and origin context fields in a message.	Yes	Yes	Yes	No	No	No	No	No	No	No

Authority	Description	Queue manager	Remote queue manager	Queue	Process definitions	Namespace	Authentication information	Channel	Client-connection channel	Service	Listener
Set identity context	Allow the application to set the identity context fields in a message, and allow the queue manager to generate the origin context.	Yes	Yes	Yes	No	No	No	No	No	No	No
System	Gives authority to principals or groups who are authorized to carry privileged operations on objects	Yes	No	No	No	No	No	No	No	No	No

Related tasks

[“Granting authorities on a queue manager” on page 168](#)

To perform an operation on a queue manager, the user must have authority to perform that particular operation on the queue manager.

[“Granting authorities on a specific object” on page 169](#)

A user must have the correct authorities to perform operations on objects; for example, to browse the messages on a queue.

[“Granting authorities on multiple objects” on page 170](#)

A user must have the correct authorities to perform operations on objects; for example, to browse the messages on a queue. You can grant the same set of authorities to multiple objects on a queue manager by using generic profiles.

[“Granting the Create authority” on page 166](#)

To create a new object on a queue manager, the user who performs the operation must have authority to create that type of object on the queue manager.

Wildcards used in generic profiles

You can use some wildcard characters in generic profiles.

The following table lists the wildcard characters that you can use in generic profiles.

Wildcard character	Description	Example
?	Use the question mark (?) instead of any single character.	AB . ?D applies to the objects AB . CD, AB . ED, and AB . FD .
*	Use the asterisk (*) as a qualifier in a profile name to match any one qualifier in an object name. A qualifier is the part of an object name delimited by a period. For example, in ABC . DEF . GHI, the qualifiers are ABC, DEF, and GHI.	ABC . * . JKL applies to the objects ABC . DEF . JKL, and ABC . GHI . JKL; it does not apply to ABC . JKL because * used in this context always indicates exactly one qualifier.
	Use the asterisk (*) as a character within a qualifier in a profile name to match zero or more characters within the qualifier in an object name.	ABC . DE* . JKL applies to the objects ABC . DE . JKL, ABC . DEF . JKL, and ABC . DEGH . JKL.
**	Use the double asterisk (**) once in a profile name as the entire profile name to match all object names.	If you use ** as the profile name, the profile applies to all processes.
	Use the double asterisk (**) once in a profile name as either the beginning, middle, or ending qualifier in a profile name to match zero or more qualifiers in an object name.	** . ABC identifies all objects with the final qualifier ABC.

Note that wildcard characters must use quotation marks on systems that expand them. In general, Linux and UNIX platforms require double quotation marks around generic profiles, whereas Windows platforms do not.

For other platforms, refer to your product documentation.

Related concepts

[“Generic and specific profiles” on page 180](#)

When you manage authorities for a folder of objects (for example, the Queues folder) using the Manage Authority Records dialog, you grant authorities against profiles instead of granting authorities on specific objects.

Related tasks

[“Granting authorities on multiple objects” on page 170](#)

A user must have the correct authorities to perform operations on objects; for example, to browse the messages on a queue. You can grant the same set of authorities to multiple objects on a queue manager by using generic profiles.

Exporting authorities to a file

You can export object authorities to a text file from IBM MQ Explorer.

About this task

The authorities are formatted in the text file so that you can use lines from the file on the command line or in scripts to set authorities on other computers in your IBM MQ network. For example, a file could contain the following lines:

```
setmqaut -m QM_A -n Q1 -t queue -p user@domain +browse +chg +clr +dlt +dsp +put +inq +get  
+passall +passid +set +setall +setid  
setmqaut -m QM_A -n Q1 -t queue -g mqm +browse +chg +clr +dlt +dsp +put +inq +get +passall  
+passid +set +setall +setid
```

You can export different subsets of object authorities:

- [“Exporting all object authorities for a queue manager and its objects” on page 189](#)
- [“Exporting all Create authorities for a queue manager” on page 189](#)
- [“Exporting authorities by object type” on page 190](#)

Exporting all object authorities for a queue manager and its objects

About this task

To export all of the object authorities for a queue manager and the queue manager's objects:

Procedure

1. In the Navigator view, right-click the queue manager, then click **Object Authorities > Save All**. A dialog opens.
2. Type a name for the text file and save the authorities.

Results

All of the object authorities for the queue manager and its objects are saved in the text file.

Exporting all Create authorities for a queue manager

About this task

To export all of the Create authorities for a queue manager:

Procedure

1. In the Navigator view, right-click the queue manager, then click **Object Authorities > Manage the Create Authorities**.
The Manage Create Authorities dialog opens. For more information about the managing Create authorities, see [Granting the Create authority](#).
2. Click **Save As**.
A dialog opens.
3. Type a name for the text file and save the authorities.

Results

All of the Create authorities for the queue manager are saved in the text file.

Exporting authorities by object type

About this task

To export all of the object authorities by object type:

All of the object authorities from the records that were found are saved in the text file.

Procedure

1. In the Navigator view, right-click the queue manager, then click **Object Authorities > Find Authorities**. The Find Authorities dialog opens.
2. Enter the search parameters as required, then click **Find**; for more information, see [Finding the authorities of a user or group](#).
3. Click **Save As A** dialog opens.
4. Type a name for the text file and save the authorities.

Related tasks

[“Exporting and importing settings” on page 258](#)

You can export your settings from IBM MQ Explorer for backup purposes, or to transfer and import the settings into another instance of IBM MQ Explorer.

[“Finding the authorities of a user or group on an object” on page 175](#)

You can search the authorization service for authority records or accumulated authorities that have been granted to groups or users (entities) on a queue manager's objects. If the group or user does not have an authority record on the specified objects, no results are displayed.

[“Granting the Create authority” on page 166](#)

To create a new object on a queue manager, the user who performs the operation must have authority to create that type of object on the queue manager.

Configuring a default security exit

A security exit can be defined for all client connections in the same IBM MQ Explorer. This is known as a default security exit.

About this task

The default security definitions will be persisted in IBM MQ Explorer and will be automatically included in **Preferences** in any import actions or export actions. The security exit details for each queue manager will be persisted with the other connection details of the queue manager.

To configure the default security exit:

Procedure

1. Click **Window > Preferences**. The **Preferences** dialog opens.
2. Expand **MQ Explorer**.
3. Expand **Client Connections**. The default security settings dialogs are now accessible.
4. Configure the security settings as required.

What to do next

The default security exit has now been configured. All new client connections in the same IBM MQ Explorer now use the settings you have configured as a default. The settings can be overridden when adding a new remote queue manager.

Related tasks

[“Configuring the client security details for a queue manager set” on page 191](#)

The client security details and security exit can be defined for all the client-connected queue managers in a queue manager set.

Related reference

[“Default security preferences” on page 192](#)

A security exit can be defined for all client connections in the same IBM MQ Explorer. This is known as a default security exit and the preferences for the security exit are described here.

[“Passwords preferences” on page 194](#)

You can store passwords to a file so that you do not have to enter them every time you want to connect to resources.

Configuring the client security details for a queue manager set

The client security details and security exit can be defined for all the client-connected queue managers in a queue manager set.

Before you begin

Before you set the security details for a queue manager set, queue manager sets must be visible, as described in: [“Displaying queue manager sets” on page 233](#).

About this task

The security definitions are persisted in IBM MQ Explorer and are automatically included in **Preferences** in any import actions or export actions. The security details for each queue manager are persisted with the other connection details of the queue manager. The security details can be set for the **All** queue manager set as well as for user-defined queue manager sets.

To configure the security details for all existing queue managers in a queue manager set:

Procedure

1. Right-click the queue manager set you want to define the security details for.
2. Click **Edit Security Settings...**
The **Set Connection Details** wizard opens, and you can set the security exit details, user ID and password details, TLS certificate store details, and enable the default TLS options. User ID and password details are also applicable to any local queue managers that are part of the set.
3. Select the security options that you want from each page of the wizard.
4. Select the queue managers that you want to apply the new security settings to. Click **Finish** to apply the changes and close the **Set Connection Details** dialog.

What to do next

The security details are configured for the selected queue manager set. All the queue managers that you selected in the queue manager set are configured with the new security details. The security configuration applies to all instances of the same queue managers in different queue manager sets.

The changes will not be applied until the next time the queue manager is connected.

Related tasks

[“Configuring a default security exit” on page 190](#)

A security exit can be defined for all client connections in the same IBM MQ Explorer. This is known as a default security exit.

Related reference

[“Default security preferences” on page 192](#)

A security exit can be defined for all client connections in the same IBM MQ Explorer. This is known as a default security exit and the preferences for the security exit are described here.

[“Passwords preferences” on page 194](#)

You can store passwords to a file so that you do not have to enter them every time you want to connect to resources.

Default security preferences

A security exit can be defined for all client connections in the same IBM MQ Explorer. This is known as a default security exit and the preferences for the security exit are described here.

The default security preferences are part of the **Preferences** dialog, and they can be opened in the following way:

1. Click **Windows > Preferences....** The **Preferences** dialog opens.
2. Expand **MQ Explorer**.
3. Expand **Client Connections**. The default security settings dialogs are now accessible.

Security Exit

Select **Enable default security exit** to set the default security exit for all client connections in the same IBM MQ Explorer. The security exit for all the client-connected queue managers in a set can be changed. The security exit can be overridden if you define a new security exit when you add a new remote queue manager.

The Security Exit for all client-connected queue managers in a set can be changed. The TLS options can be overridden when you add a new remote queue manager.

Item	Description
Exit name	Specifies the name of the exit program to be run by the security exit. Exit name can be up to 1024 characters long and is case sensitive. Exit name can be a fully qualified java class name found in the directory or jar file. Exit name can be a C exit, of the format: <code>dll_name(function_name)</code> . The default path for exits is always used to locate C exits, you cannot specify the location of the exit library in this entry field unless no default path is set.
in directory	Specifies the directory for the security exit (Java exits only).
in jar	Specifies the jar file for the security exit (Java exits only).
Exit data	Exit data can be up to 32 characters long. If no value has been defined for that attribute, this field is all blanks.

SSL/TLS Options

Select **Enable default SSL options** to enable the default SSL/TLS options for all client connections in the same IBM MQ Explorer. The SSL/TLS options for all client-connected queue managers in a set can be changed. The SSL/TLS options can be overridden when you add a new remote queue manager.

Item	Description
SSL CipherSpec	<p>The CipherSpec identifies the combination of encryption algorithm and hash function used by an SSL/TLS connection. A CipherSpec forms part of a CipherSuite, which identifies the key exchange and authentication mechanism as well as the encryption and hash function algorithms.</p> <p>The size of the key used during the handshake can depend on the digital certificate you use, but some of the CipherSpecs supported by IBM MQ include a specification of the handshake key size. Note that larger handshake key sizes provide stronger authentication. With smaller key sizes, the handshake is faster.</p> <p>For more information, see CipherSpecs and CipherSuites in the IBM MQ online product documentation.</p>
SSL FIPS required	<p>Select Yes to use only FIPS-certified cipher suites. If you select Yes, then all TLS connections must use FIPS-certified cipher suites.</p> <p>Select No to use any available cipher suites.</p> <p>The default setting is No.</p> <p>If you change this setting from Yes to No, or from No to Yes a dialog will be opened asking if you want to restart MQ Explorer.</p> <p>Any changes to this setting will not be applied until the MQ Explorer has been restarted.</p>
SSL reset count	<p>Type the number of bytes, from 0 to 999 999 999, that are sent and received within a TLS conversation before the secret key is renegotiated. A value of 0 means that the secret key is never renegotiated. The number of bytes includes control information that is sent by the message channel agent (MCA). If the value of this attribute is greater than 0 and the value of the Heartbeat interval attribute in the Channel properties is greater than 0, the secret key is also renegotiated before message data is sent or received following a channel heartbeat.</p>
Peer name	<p>The Distinguished Name (DN) of the queue manager to be used by TLS. The peer name is set to indicate that connections will only be allowed where the server is successfully authenticated as a specific DN.</p>

SSL/TLS Stores

Select **Enable default SSL stores** to work with the Trusted Certificate Store and the Personal Certificate Store.

To configure IBM MQ Explorer with the location and password of the SSL/TLS certificate store, refer to: [“Specifying the default location and default password of TLS certificates” on page 105.](#)

By enabling the default SSL/TLS stores, IBM MQ Explorer can use the certificates in the TrustStore and KeyStore to connect to remote queue managers with a TLS-enabled connection.

The SSL/TLS Stores for all client-connected queue managers in a set can be changed. The SSL/TLS Stores can be overridden when you add a new remote queue manager.

Related tasks

[“Configuring a default security exit” on page 190](#)

A security exit can be defined for all client connections in the same IBM MQ Explorer. This is known as a default security exit.

[“Configuring the client security details for a queue manager set” on page 191](#)

The client security details and security exit can be defined for all the client-connected queue managers in a queue manager set.

Related reference

[“Passwords preferences” on page 194](#)

You can store passwords to a file so that you do not have to enter them every time you want to connect to resources.

Passwords preferences

You can store passwords to a file so that you do not have to enter them every time you want to connect to resources.

Passwords used by the IBM MQ Explorer to connect to resources (for example: opening TLS stores or connecting to queue managers), can be stored in a file. The password file can be stored locally, to a remote device, or to a removable device.

To open the **Passwords** preference panel:

1. Click **Window > Preferences**. The **Preferences** dialog opens.
2. Expand **MQ Explorer**.
3. Select **Passwords** to display the **Passwords** panel.

Item	Description
Do not save passwords	Passwords are not stored to a file. This is the default value.
Save passwords to file	Passwords are saved to the file you specify. Select Save passwords to file and click Browse to select a location for the encrypted password file
Use default key	You must use a key to open a password store. This is the default value.
User defined key	You must use a key to open a password store. Select User defined key then click Change to enter your password. The password must contain a minimum of 8 characters.

Related tasks

[“Configuring a default security exit” on page 190](#)

A security exit can be defined for all client connections in the same IBM MQ Explorer. This is known as a default security exit.

[“Configuring the client security details for a queue manager set” on page 191](#)

The client security details and security exit can be defined for all the client-connected queue managers in a queue manager set.

Related reference

[“Default security preferences” on page 192](#)

A security exit can be defined for all client connections in the same IBM MQ Explorer. This is known as a default security exit and the preferences for the security exit are described here.

Configuring API exits

An API exit is a code module, a .dll file, that you provide yourself and that runs immediately before or after MQI calls.

About this task

When IBM MQ receives a call from a program to one of its API entry points, IBM MQ calls your API exit. The API exit runs either before or after the MQI is run, according to how you configured the exit.

You can configure none, one, or many exits to be called, and you can configure the sequence in which multiple exits are called. On Windows and Linux (x86 and x86-64 platforms), you can configure the API exits using IBM MQ Explorer. The configuration details are stored in .ini files.

There are three types of API exit definition:

Common (ApiExitCommon)

One set of definitions per computer. When the queue manager starts, the API exits that are defined, if any, are read and applied to the queue manager. Configure common API exits in the IBM MQ properties dialog. Common exits are displayed in the **Local API Exits** table in the properties dialog of each local queue manager.

Template (ApiExitTemplate)

One set of definitions per computer. When a queue manager is created, the API exits defined here, if any, are copied into the newly created queue manager as local exits. Configure template API exits in the IBM MQ properties dialog.

Local (ApiExitLocal)

One set of definitions per queue manager. When the queue manager starts, any API exits that are defined override the common exits if their **Name** attributes are the same, and if the override has been specified. When a common API exit is overridden, none of the fields in the common definition are saved, even if the optional **Data** attribute has an assigned value. Configure local API exits in the queue manager's properties dialog.

Configuring an API exit in IBM MQ Explorer

About this task

To configure an API exit:

Procedure

1. Open the relevant properties dialog:
2. On the **Exits** page, click **Add...**. The **Add API Exit** dialog opens.
3. Type the required information into the fields of the **Add API Exit** dialog.
4. Click **OK** to create the exit and close the **Add API Exit** dialog.

Results

The properties of the new API exit are displayed in the table on the Exits page.

Overriding a common API exit with a local API exit

About this task

When a local API exit is defined on a queue manager with the same name as a common exit, the common exit is overridden. That is, the common exit is not called; instead, the overriding local exit is called. To prevent accidental overriding, the user interface makes you take deliberate actions to configure an override; for example, you cannot add a new exit with the same name as an existing exit, and you cannot change the name of an exit to be the same as an existing exit.

However, you might want to add a local API exit to a queue manager so that the common API exit is not used and the local API exit is used instead. In this case, you need to override the common API exit with the local API exit.

To override a common API exit with a local API exit:

Procedure

1. Open the **Exits** page of the queue manager properties dialog.
2. Click the common exit that you want to override in the **Local API Exits** table.
3. Click **Override**.
The **Edit API Exit** dialog opens with the name of the common API exit displayed.
4. Type the details of the local API exit in the **Edit API Exit** dialog, and click **OK** to save the changes.

Results

The local exit now overrides the common exit that has the same name.

API exit attributes

About this task

When you configure API exits in the IBM MQ and queue manager properties dialogs, the attribute values are added to the ApiExitCommon, ApiExitTemplate, and ApiExitLocal stanzas in the configuration files or the Windows registry.

Attribute	Meaning	Stanza key
Name	Specifies the descriptive name of the API exit which is passed to the API exit in the ExitInfoName field of the MQAXP structure. This name must be unique and is limited to a length of 48 characters, and must contain only those characters that are valid for the name of IBM MQ objects, such as queue names.	Name
Type	Specifies the type of exit: common, template, local, or override.	(Not a separate stanza key.)
Sequence	This attribute is an unsigned numeric value that defines the sequence in which this API exit is called in comparison with other API exits. An API exit with a low sequence number is called before another API exit with a higher sequence number. The order in which different API exits with the same sequence number are called is undefined. It is perfectly valid for there to be gaps in the sequence numbers of the API exits defined for a queue manager.	Sequence
Module	Specifies the module that contains the code for the API exit. If this field contains the full path name of the module it is used as is. If this field contains just the module name, the module is located using the same method as channel exits; that is, using the value in the Exit default path field on the Exits page of the queue manager properties dialog.	Module
Function	Specifies the name of the function entry-point into the module that contains the code for the API exit. This entry-point is the MQ_INIT_EXIT function. The length of this field is limited to MQ_EXIT_NAME_LENGTH.	Function

Attribute	Meaning	Stanza key
Data	If this attribute is specified, leading and trailing blanks are removed, the remaining string is truncated to 32 characters, and the result is passed to the exit in the ExitData field of the MQAXP structure. If this attribute is not specified, the default value of 32 blanks is passed to the exit in the ExitData field of the MQAXP structure.	Data

[“Configuring queue managers and objects” on page 51](#)

You can configure many of the attributes of queue managers and their objects from IBM MQ Explorer using the properties dialogs.

[“Queue manager properties” on page 349](#)

You can set attributes for both local and remote queue managers.

[“IBM MQ properties” on page 340](#)

IBM MQ properties apply to the whole IBM MQ installation.

Authorizing users to configure IBM MQ on Windows and Linux (x86 and x86-64 platforms)

IBM MQ uses the normal user and group authorizations to protect IBM MQ applications and IBM MQ administration.

Configuring IBM MQ

About this task

The IBM MQ installation automatically creates the local group mqm. Only users that belong to the mqm group can perform tasks such as creating, deleting, and altering queue managers, setting authorizations on queue manager objects, and running listeners. For more information regarding the commands that are used to perform these tasks, see [IBM MQ control commands reference](#) in the IBM MQ online product documentation..

On Windows, user names that are members of the Windows Administrators group also have the authority to perform these tasks. Users that are members of the Windows Administrators group are also authorized to alter the local Windows operating system settings. For IBM MQ on Windows, user names can contain a maximum of 20 characters; for IBM MQ on other platforms, user names can contain a maximum of only 12 characters.

To give a user authority to administer queue managers:

Procedure

1. Log in to the operating system with a user name that has Administrator authority on Windows, or root authority on Linux.
2. Add the user's user name to the mqm group.

Results

On Windows, the security token that the IBM MQ Explorer queries for authority when it starts, contains the user name and authority information and is cached by Windows. If changes are made to a user name authorization, that user must log off and on again for the changes to take effect when IBM MQ Explorer is restarted.

Performing IBM MQ operations

About this task

To perform operations such as connecting to a queue manager, opening a queue, or creating a queue, the user must have the correct IBM MQ privileges. Only users who belong to the mqm group or who have been granted **+chg** permission on the queue manager can perform tasks such as creating, deleting, and altering queue managers. A user that has the correct privileges can run applications but cannot, for example, create or delete queue managers unless they are also a member of the mqm group.

You can make user name authorizations with various levels of capability for the IBM MQ applications you create and implement on your own network so that, for example, a user name might have the authority to connect to a queue manager and put and get messages to a queue, but not have authority to alter the attributes of that queue. Use the `setmqaut` command to do this. For more information, see [setmqaut](#) in the IBM MQ online product documentation. You can make the user names that use your application members of a global group for your network, and then, on each computer where the application must run, make the global group a member of the mqm group.

Changes made to IBM MQ authorizations by the `setmqaut` command take immediate effect. However, changes made to user name authorization do not take effect until the relevant queue manager is stopped and restarted.

Running the Postcard application

About this task

To run the Postcard application, as with applications that you have written yourself, the user must have the correct privileges. Otherwise, the user receives error messages from the IBM MQ API.

Starting Windows service for an IBM MQ installation

About this task

The service starts at Windows startup time, before any user is logged on. The service is used to start any queue managers configured with the automatic startup option. In order to ensure that queue manager processes run with correct authority, the service must be configured with an appropriate user name. For more information on configuring the IBM MQ service, see [Changing the password of the IBM MQ Windows service local user account](#) in the IBM MQ online product documentation.

Refreshing the authorization service information on Multiplatforms

On Multiplatforms, if you make a change to an entity, you must refresh the entity information in the authorization service. You must do this for each queue manager that is affected by the changes that you make to the entity.

About this task

The users and groups (entities) that are displayed in the authorization service are defined in the operating system. You cannot, therefore, create or delete entities from within the authorization service itself. If you make a change to an entity (either a user or a group) while the queue manager is running, you must refresh the entity information in the authorization service.

When you refresh the entity information in the authorization service, the authorization service rebuilds its Access Control List (ACL) using the new entity information.

To refresh the entity information in the queue manager's authorization service using IBM MQ Explorer, complete the following steps.

Procedure

1. In the Navigator view, right-click the queue manager for which you want to refresh the entity information, then click **Security > Refresh Authorization Service**.
2. When prompted, click **Yes**.

Results

The entity information for the queue manager and all of its objects is refreshed in the authorization service.

Ensure that you refresh the entity information for each queue manager that is affected by the changes that you made to the entity.

Related concepts

[“Users and groups \(entities\) in the authorization service” on page 179](#)

In the authorization service, authorities are granted to users (also known as principals when the user name is fully qualified with the domain name) or groups of users for accessing IBM MQ objects. Users and groups are collectively known as entities in the authorization service. You grant a set of authorities to an entity by creating an authority record.

Related tasks

[“Refreshing TLS security” on page 200](#)

You can make changes to the key repository without restarting a channel. However, the copy of the key repository that is held in memory while a channel is running will not be affected. When you refresh the cached copy of the key repository, the TLS channels that are currently running on the queue manager are updated with the new information.

[“Refreshing ESM classes \(z/OS only\)” on page 201](#)

IBM MQ for z/OS does not perform any authority checks itself; instead, it routes requests for authority checks to an external security manager (ESM).

[“Refreshing the connection authentication configuration” on page 199](#)

If the configuration for connection authentication changes, you must refresh the queue manager's view of this configuration.

Refreshing the connection authentication configuration

If the configuration for connection authentication changes, you must refresh the queue manager's view of this configuration.

About this task

When the configuration to enable or disable connection authentication or the details of the user repository to be used for connection authentication are changed, you must refresh the queue manager's view of this configuration.

Procedure

1. In the Navigator view, right-click the queue manager for which you want to refresh the connection authentication configuration, then click **Security > Refresh Connection Authentication**.
2. When prompted, click **Yes**.

Results

The configuration for connection authentication is picked up by the queue manager and will be used to determine whether connection authentication should be applied to any subsequent connections to the queue manager.

Related tasks

[“Refreshing the authorization service information on Multiplatforms” on page 198](#)

On Multiplatforms, if you make a change to an entity, you must refresh the entity information in the authorization service. You must do this for each queue manager that is affected by the changes that you make to the entity.

[“Refreshing ESM classes \(z/OS only\)” on page 201](#)

IBM MQ for z/OS does not perform any authority checks itself; instead, it routes requests for authority checks to an external security manager (ESM).

[“Refreshing TLS security” on page 200](#)

You can make changes to the key repository without restarting a channel. However, the copy of the key repository that is held in memory while a channel is running will not be affected. When you refresh the cached copy of the key repository, the TLS channels that are currently running on the queue manager are updated with the new information.

Refreshing TLS security

You can make changes to the key repository without restarting a channel. However, the copy of the key repository that is held in memory while a channel is running will not be affected. When you refresh the cached copy of the key repository, the TLS channels that are currently running on the queue manager are updated with the new information.

About this task

When a channel is secured using TLS, the digital certificates and their associated private keys are stored in the key repository. A copy of the key repository is held in memory while a channel is running. If you make a change to the key repository the changes do not become active in the in-memory copy of the key repository while a channel is running.

When you refresh the cached copy of the key repository, all TLS channels that are currently running are updated:

- Sender, server, and cluster-sender channels that use TLS are allowed to complete the current batch of messages. The channels then run the SSL handshake again with the refreshed view of the key repository.
- All other channel types that use TLS are stopped. If the partner end of the stopped channel has retry values defined, the channel retries and runs the SSL handshake again. The new SSL handshake uses the refreshed view of the contents of the key repository, the location of the LDAP server to be used for the Certificate Revocation Lists, and the location of the key repository. In the case of server-connection channel, the client application loses its connection to the queue manager and has to reconnect to continue.

To refresh the cached copy of the key repository, complete the following steps.

Procedure

1. In the Navigator view, right-click the queue manager for which you want to refresh the cached copy of the key repository, then click **Security > Refresh SSL**.
2. When prompted, click **Yes**.

Results

The TLS channels that are currently running on the queue manager are updated with the new information. The queue manager FIPS configuration (SSLFipsRequired) is also refreshed by this command on Windows, Linux and UNIX platforms.

Related tasks

[“Securing channels with TLS” on page 155](#)

The TLS (Transport Layer Security) protocol enables queue managers to communicate securely with other queue managers, or clients.

[“Refreshing the authorization service information on Multiplatforms” on page 198](#)

On Multiplatforms, if you make a change to an entity, you must refresh the entity information in the authorization service. You must do this for each queue manager that is affected by the changes that you make to the entity.

[“Refreshing ESM classes \(z/OS only\)” on page 201](#)

IBM MQ for z/OS does not perform any authority checks itself; instead, it routes requests for authority checks to an external security manager (ESM).

[“Refreshing the connection authentication configuration” on page 199](#)

If the configuration for connection authentication changes, you must refresh the queue manager's view of this configuration.

Refreshing ESM classes (z/OS only)

IBM MQ for z/OS does not perform any authority checks itself; instead, it routes requests for authority checks to an external security manager (ESM).

About this task

The IBM MQ product documentation assumes that you are using the z/OS Security Server Resource Access Control Facility (RACF®) as the ESM.

So that IBM MQ does not have to contact RACF for every authority check, IBM MQ puts information about the user and the user's authorities in a cache. When you add, delete, or change a RACF resource profile that is held in one of the following classes:

- MQADMIN
- MQNLIST
- MQPROC
- MQQUEUE
- MXADMIN
- MXNLIST
- MXPROC
- MXQUEUE
- MXTOPIC

force IBM MQ to refresh the ESM classes so that it throws away the cached information and starts to rebuild the cache from RACF.

For more information, see [Script \(MQSC\) Commands](#) and [Introduction to Programmable Command Formats](#) in the IBM MQ online product documentation.

To refresh z/OS classes:

Procedure

1. In the Navigator view, right-click the queue manager for which you want to refresh the classes, then, to refresh all of the classes, click **Security > Refresh ESM Classes > ALL**. Alternatively, instead of clicking **ALL**, click the type of class that you want to refresh:
2. When prompted, click **Yes**.

Results

The classes that you selected are refreshed: the profiles are deleted from the in-storage table and must be retrieved directly from RACF next time they are needed.

Related tasks

[“Refreshing the authorization service information on Multiplatforms” on page 198](#)

On Multiplatforms, if you make a change to an entity, you must refresh the entity information in the authorization service. You must do this for each queue manager that is affected by the changes that you make to the entity.

[“Refreshing TLS security” on page 200](#)

You can make changes to the key repository without restarting a channel. However, the copy of the key repository that is held in memory while a channel is running will not be affected. When you refresh the cached copy of the key repository, the TLS channels that are currently running on the queue manager are updated with the new information.

[“Refreshing the connection authentication configuration” on page 199](#)

If the configuration for connection authentication changes, you must refresh the queue manager's view of this configuration.

Viewing the status of objects

You can display the current status of any object that can be in different states, in IBM MQ Explorer. For IBM MQ channels, you can also view the saved status.

About this task

To view the status of an object in IBM MQ Explorer, complete the following steps.

Procedure

In the **Content** view, right-click the object, then click **Status...**

If you are viewing a channel definition's status, you can view the current status of the channel (click **Channel Status**) or the saved status of the channel (click **Saved Status**).

Results

The Status dialog for the object opens displaying the status information that you requested.

To view the status of all the objects of a specific type, for a selected queue manager:

In the **Explorer** view, right-click the folder of objects (for example Queues) for a selected queue manager, then click **Status....** A new **Content** view will be opened in a separate window. The status of all the objects in the object-folder are displayed in the new **Content** view window.

Viewing the status of multiple instances of the same receiver channel

About this task

Different applications can use different instances of the same receiver channel at the same time. It is possible for these different instances to have different statuses.

There are two ways to view the status of multiple channel instances in the IBM MQ Explorer. The first way is:

- In the **Content** view, right-click the channel, then click **Status...** You can view the current status of the channel (click **Channel Status**) or the saved status of the channel (click **Saved Status**). All the statuses for the individual instances are aggregated into a single status displayed in the **Content** view.

The second way is:

- In the **Navigator** view, right-click the channels folder of your selected queue manager, then click **Status**. You can view the current status of the channel (click **Channel Status**) or the saved status of the channel (click **Saved Status**). A new **Content** view will be opened in a separate window. The status of all the objects in the folder are displayed in the new **Content** view window. All of the channel instances and the individual statuses are displayed in the Content view.

Results

The aggregated status displayed is dependant on the number of instances and their different statuses, as described in the following points:

- There are no channel instances: Status shown as **Inactive**.
- There is a single channel instance: Status shown as the actual status of the channel.
- There are more than 1 instances, all with the same status: Status shown as the actual status of the channels.
- There are more than 1 instances, with mixed statuses: Status shown as **Mixed**.

Related tasks

[“Configuring queue managers and objects” on page 51](#)

You can configure many of the attributes of queue managers and their objects from IBM MQ Explorer using the properties dialogs.

Related reference

[“Status attributes” on page 604](#)

In IBM MQ Explorer, you can view the current status of IBM MQ objects. For example, you can find out whether a channel is running, or you can find out when the last message was put on a certain queue. You can also view the saved status of a channel.

Viewing and closing connections to applications

You can use the **Application Connections** dialog to find out which applications are currently connected to a specific queue manager, and which queue manager objects an application is currently accessing. You can also use this dialog to close a connection.

About this task

Before you delete an IBM MQ object, or change its attributes, check whether any applications are currently connected to the queue manager or accessing its objects. The **Application Connections** dialog displays the applications that are currently connected to a specific queue manager, and which queue manager objects the application is currently accessing.

You can use the **Application Connections** dialog to close a connection. Be aware that when you close a connection between an application and a queue manager, the application can no longer access any of the queue manager's objects. This might prevent the application from working correctly.

To view a list of the applications that are connected to a queue manager:

Procedure

1. In the **Navigator** view, right-click the queue manager, then click **Application Connections**. The **Application Connections** dialog opens.
2. In the **Application Connections** dialog, the first table lists the applications that are currently connected to the queue manager.
3. Click an application to display, in the second table, a list of the objects on the queue manager that the application is accessing.
4. Optional: Close a connection:
 - a) Click the name of the application, then click **Close Connection**.
 - b) When you are prompted, click **Yes** to confirm that you want to close the connection.The connection between the application and the queue manager is closed.

Results

If you closed a connection, the application that used that connection can no longer access the queue manager's objects.

Creating and configuring JMS administered objects

You can use IBM MQ Explorer to configure the JMS administered objects that enable communication between Java applications and IBM MQ.

About this task

The Java Message Service (JMS) is a Java API that allows applications that are written in Java to the JMS specification to communicate with any messaging product that implements the JMS API. Because the JMS API is an open standard and has multiple implementations, you can select which messaging provider (the JMS provider) to use according to your requirements.

IBM MQ is a JMS provider; it provides a messaging system that implements the JMS API. You can use the IBM MQ graphical user interface, IBM MQ Explorer, to configure the JMS administered objects that enable communication between the Java application (the JMS client) and IBM MQ (the JMS provider).

There are two types of JMS administered objects in IBM MQ classes for JMS:

- [Connection factories](#), which the JMS client uses to create connections to the JMS provider.
- [Destinations](#), which the JMS client uses to represent the target and source of messages.

The administered objects are stored in a naming and directory service that is accessed by IBM MQ Explorer using the Java Naming Directory Interface (JNDI) API. The administered objects are stored in locations on the naming and directory service known as the JNDI namespaces. There are various JNDI service providers that you can use as the naming and directory service, including Lightweight Directory Access Protocol (LDAP) and local or remote file systems.

Because the JMS client uses the JMS administered objects to connect to the JMS provider, you configure the administered objects to define how the JMS client sends and receives, or publishes and subscribes, messages. The JMS client does not directly communicate with the JMS provider and is unaware of which JMS provider is being used. This means that you can change the JMS provider without having to update the JMS client.

To configure IBM MQ classes for JMS so that a JMS client can connect to and access administered objects in a JNDI namespace, you must perform the following tasks in IBM MQ Explorer:

Procedure

1. Connect to the JNDI namespace. For more information, see [Adding an initial context](#).
2. Create and configure the administered objects that are stored in the JNDI namespace. For more information, see [Creating a connection factory](#) and [Creating a destination](#).

Results

For more information about programming JMS applications and configuring IBM MQ classes for JMS, see [Using IBM MQ classes for JMS](#) in the IBM MQ online product documentation.

Related concepts

[“JMS connection factories” on page 206](#)

A connection factory is an object that a JMS client (a JMS program that uses the JMS API) uses to create a connection with a JNDI provider (a messaging provider such as IBM MQ).

[“JMS destinations \(queues and topics\)” on page 208](#)

A JMS destination is an object (a JMS queue or a JMS topic) that represents the target of messages that the client produces and the source of messages that the client consumes. In point-to-point messaging, destinations represent queues; in publish/subscribe messaging, destinations represent topics.

JMS contexts

A context is a set of bindings that associates names with objects stored in a naming and directory service.

JMS clients (Java applications that use the JMS API) use contexts to look up the names of the JMS objects in the naming and directory service. Every context has a naming convention associated with it.

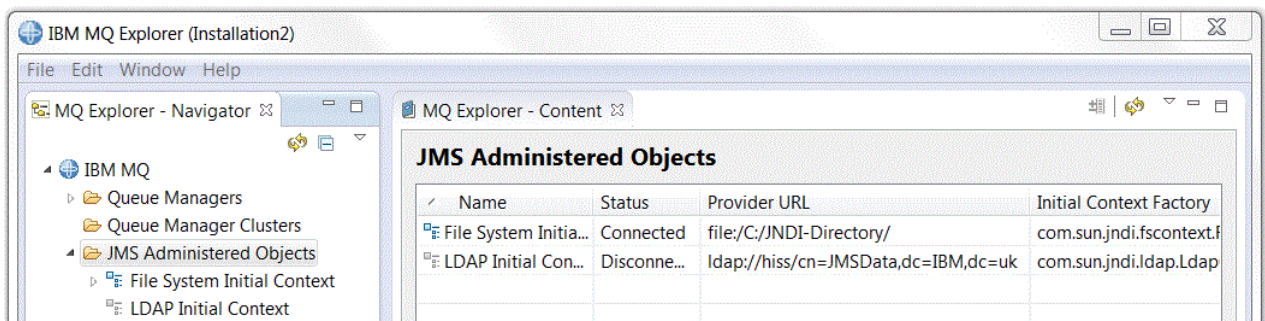
For more information about LDAP naming considerations, see [Configuring the JMS administration tool](#) in the online product documentation.

Initial contexts

For each location in the naming and directory service, you need to specify an initial context to give a starting point from which the JMS client can resolve the names of the objects in that location of the naming and directory service. JMS clients access the objects in the naming and directory service through the Java Naming Directory Interface (JNDI); the location in the naming and directory service that is defined by the context is known as the JNDI namespace.

When you specify an initial context in IBM MQ Explorer, the full contents of the JNDI namespace are displayed but, in IBM MQ Explorer, you can edit only the IBM MQ classes for JMS objects that are stored there. All of the initial contexts that you add to IBM MQ Explorer are displayed in the **Navigator** view in the **JMS Administered Objects** folder, as shown in the following figure.

In the figure, **File System Initial Context** is the initial context for a location in the local filesystem: `C:/JMSAdmin/JMSAdmin1` and **LDAP Initial Context** is the initial context for a location on an LDAP server, on a computer called `hiss` with the distinguished name `cn=JMSData,dc=ibm,dc=uk`.

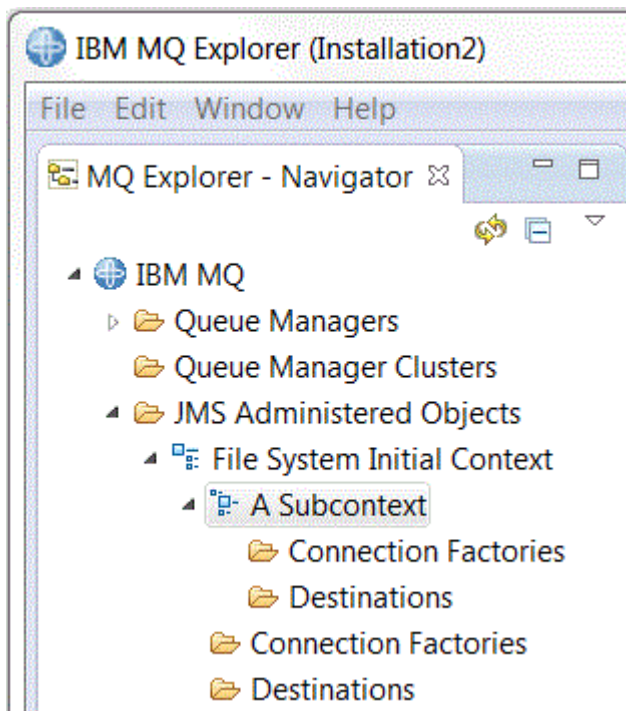


When you have added the initial context to IBM MQ Explorer, you can create connection factory objects, destination objects, and subcontexts in the JNDI namespace.

Subcontexts

A subcontext is a subdivision of a JNDI namespace and can contain connection factories and destinations as well as other subcontexts. A subcontext is not an object in its own right; it is merely an extension of the naming convention for the objects in the subcontext. You can create multiple subcontexts in a single context.

In the following figure, the subcontext called **A Subcontext** is bound to the initial context called **File System Initial Context**. In the file system where the context and subcontext are stored, the subcontext is a sub-directory of the initial context; other JNDI implementations, such as LDAP, might store subcontexts differently.



You can create connection factory objects, destination objects, and other subcontexts in a subcontext.

Related concepts

[“JMS connection factories” on page 206](#)

A connection factory is an object that a JMS client (a JMS program that uses the JMS API) uses to create a connection with a JNDI provider (a messaging provider such as IBM MQ).

[“JMS destinations \(queues and topics\)” on page 208](#)

A JMS destination is an object (a JMS queue or a JMS topic) that represents the target of messages that the client produces and the source of messages that the client consumes. In point-to-point messaging, destinations represent queues; in publish/subscribe messaging, destinations represent topics.

Related tasks

[“Adding an initial context” on page 209](#)

To create and configure JMS objects in IBM MQ Explorer, you must add an initial context to define the root of the JNDI namespace in which the JMS objects are stored in the naming and directory service.

[“Connecting and disconnecting an initial context” on page 211](#)

You can connect or disconnect IBM MQ Explorer to an initial context that is displayed in the JMS Administered Objects folder. You can also configure each initial context so that IBM MQ Explorer automatically reconnects to it the next time that you close and restart IBM MQ Explorer.

[“Creating a subcontext” on page 220](#)

A subcontext is a subdivision of a JNDI namespace and can contain connection factories and destinations as well as other subcontexts. You can create subcontexts within initial contexts or within other subcontexts.

Related information

<https://java.sun.com/products/jndi/tutorial/TOC.html>

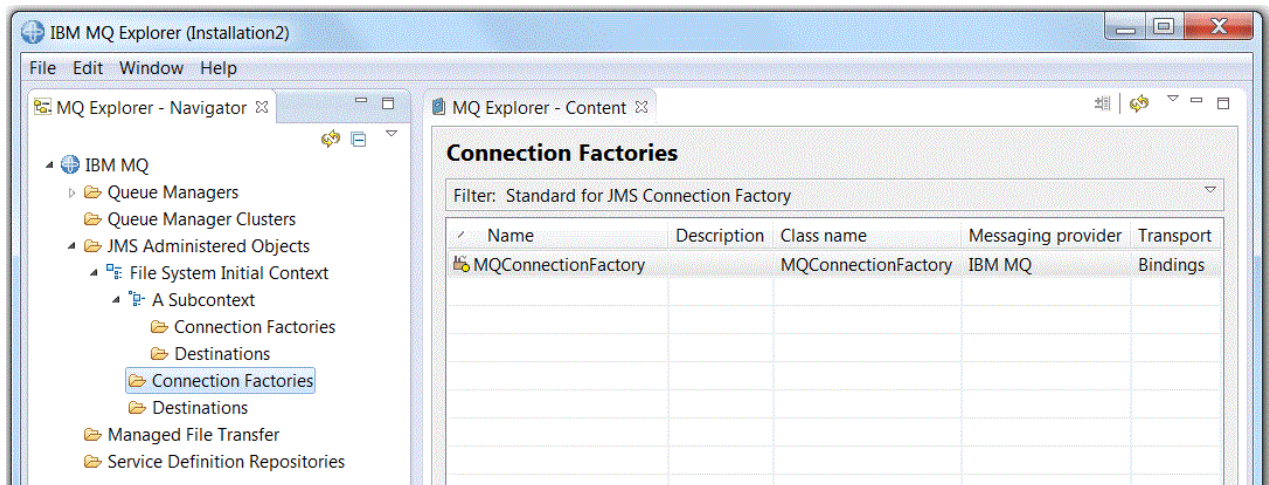
JMS connection factories

A connection factory is an object that a JMS client (a JMS program that uses the JMS API) uses to create a connection with a JNDI provider (a messaging provider such as IBM MQ).

You can use IBM MQ Explorer to create connection factories and to define the connection parameters that the connection factory will use to create connections.

Connection factories, like destinations, are administered objects and are stored in a JNDI namespace, which is a defined location within the naming and directory service. The initial context defines the root of the JNDI namespace. In IBM MQ Explorer, all connection factories are stored in Connection Factories folders in the appropriate context and subcontexts, as shown in the following figure.

In the figure, the connection factory that is called Connection Factory 1 is stored in the Connection Factories folder of the initial context called File System Initial Context.



When you define a connection factory, you select the messaging provider that is used as the JMS provider (for example, IBM MQ or Real-time); a connection factory can create connections only to that messaging provider. For the JMS client to create connections to a different messaging provider, you must create a new connection factory and specify the messaging provider. Real-time transport is not available in IBM MQ 8.0. If you are using IBM MQ 8.0 you can define Real-time transport, but it fails when an attempt is made to create a connection.

Domain-independent connection factories

There are two messaging domains: the point-to-point messaging domain and the publish/subscribe messaging domain. You can create a connection factory to create connections specifically for point-to-point messaging (using the `QueueConnectionFactory` interface) or specifically for publish/subscribe messaging (using the `TopicConnectionFactory` interface). From JNDI, you can also create connection factories that are domain independent and so can be used for both point-to-point and publish/subscribe messaging (using the `ConnectionFactory` interface). For more information, see [Creating a connection factory](#).

If the JMS application is intended to use only point-to-point messaging or only publish/subscribe messaging, you can select the specific messaging domain when you create the connection factory and a domain-specific connection factory is created.

If, however, you want to perform both point-to-point and publish/subscribe work under the same transaction, you can create a domain-independent connection factory. For example, you might want a JMS application to subscribe to a topic (publish/subscribe messaging) then when the JMS application receives a particular message, it sends another message to a queue (point-to-point messaging). It is difficult to reliably perform both point-to-point and publish/subscribe work under the same transaction if you use domain-specific connection factories: you must create a separate connection factory for each messaging domain, which means that the point-to-point work is done under a transaction that is controlled by the `QueueSession` session and the publish/subscribe work is done under a transaction that is controlled by the `TopicSession` session. It is difficult to ensure that the send and receive actions either both happened or both were backed out.

Instead of creating one domain-specific connection factory for the point-to-point work and one domain-specific connection factory for the publish/subscribe work, you can create a single domain-independent connection factory for both. This means that the connection factory creates one connection, which creates one session. The session creates a `MessageConsumer` from a topic and a `MessageProducer` to a

queue. When the JMS application receives the published message, the next message can be sent to the queue under the same session's transaction; both operations can then be committed or rolled back as a single unit of work.

For more information, see [Using IBM MQ classes for JMS](#) in the IBM MQ online product documentation.

Related concepts

[“JMS destinations \(queues and topics\)” on page 208](#)

A JMS destination is an object (a JMS queue or a JMS topic) that represents the target of messages that the client produces and the source of messages that the client consumes. In point-to-point messaging, destinations represent queues; in publish/subscribe messaging, destinations represent topics.

Related tasks

[“Creating a connection factory” on page 213](#)

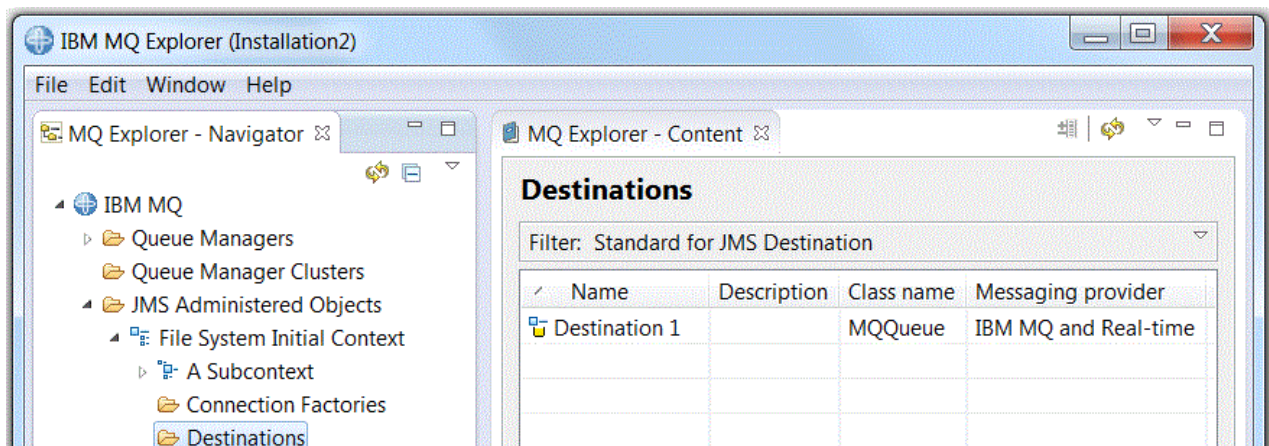
A JMS client (a Java application that uses the JMS API) uses connection factories to create connections to the JMS provider (a messaging provider such as IBM MQ).

JMS destinations (queues and topics)

A JMS destination is an object (a JMS queue or a JMS topic) that represents the target of messages that the client produces and the source of messages that the client consumes. In point-to-point messaging, destinations represent queues; in publish/subscribe messaging, destinations represent topics.

The JMS client can use a single destination object to put messages on and to get messages from, or the JMS client can use separate destination objects. The same destination object can be used for both IBM MQ and Real-time messaging providers so, unlike with connection factories, you do not need to create separate destination objects for different messaging providers. Real-time transport is not available in IBM MQ 8.0. If you are using IBM MQ 8.0 you can define Real-time transport, but it fails when an attempt is made to create a connection.

In the figure, the destination called `Destination 1` is stored in the `Destinations` folder of the initial context called `File System Initial Context`.



When you create a destination object, you must specify whether the destination is a JMS queue (in the point-to-point messaging domain) or a JMS topic (in the publish/subscribe messaging domain); you cannot change the domain after the destination has been created. You must also configure the destination with the name of the queue or topic that the destination represents. An advantage of using JMS is that you can change the name of the queue or topic that the JMS client uses by changing the value of a property in the destination definition and you do not update the JMS client itself.

For more information, see [Using IBM MQ classes for JMS](#) and [Publish/subscribe messaging](#) in the IBM MQ online product documentation.

Related concepts

[“IBM MQ queues” on page 29](#)

A queue is a container for messages. Business applications that are connected to the queue manager that hosts the queue can retrieve messages from the queue or can put messages on the queue.

Messaging providers for IBM MQ classes for JMS

A JMS client (a Java application that uses the JMS API) uses a connection factory to create a connection with the JMS provider. The messaging provider that is used as the JMS provider determines which types of transport are available to use for the connection.

If you are using point-to-point messaging or if you are using the IBM MQ Publish/Subscribe broker for publish/subscribe messaging, you must use IBM MQ as the messaging provider. The JMS client, therefore, connects to a queue manager and the type of transport that the connection uses depends on whether the JMS client is on the same computer as the queue manager:

- If the JMS client is on a different computer from the queue manager, the JMS client must use a client connection (TCP/IP) to connect to the queue manager.
- If the JMS client is on the same computer as the queue manager, the JMS client can connect to the queue manager directly using bindings or using a client connection (TCP/IP).

When you create a connection factory, you select which messaging provider will be used as the JMS provider. This constrains the connection factory to creating connections suitable only for the selected messaging provider. If you want to change messaging provider, you must create a new connection factory and specify a different messaging provider. For more information, see [Creating a connection factory](#).

You can, however, change the type of transport used as long as the new transport is appropriate for the selected messaging provider. To do this, you must change the transport type that is associated with the connection factory that the JMS client uses to create connections. For more information, see [Changing the transport type used for connections](#).

Related concepts

[“JMS connection factories” on page 206](#)

A connection factory is an object that a JMS client (a JMS program that uses the JMS API) uses to create a connection with a JNDI provider (a messaging provider such as IBM MQ).

Related tasks

[“Creating a connection factory” on page 213](#)

A JMS client (a Java application that uses the JMS API) uses connection factories to create connections to the JMS provider (a messaging provider such as IBM MQ).

[“Changing the transport type used for connections” on page 219](#)

You can change the transport type that a JMS client uses to connect to a JMS provider. You might also need to change any properties and settings that are required by the new transport type.

Adding an initial context

To create and configure JMS objects in IBM MQ Explorer, you must add an initial context to define the root of the JNDI namespace in which the JMS objects are stored in the naming and directory service.

About this task

You must add an initial context for each JNDI namespace that you want to access. All of the initial contexts that you add to IBM MQ Explorer are displayed in the JMS Administered Objects folder in the Navigator view.

To add an initial context to the JMS Administered Objects folder in IBM MQ Explorer, complete the following steps.

Procedure

1. In the Navigator view, right-click the **JMS Administered Objects** folder, then click **Add Initial Context**. The Add Initial Context wizard opens.

2. Select the location of the JNDI namespace in the naming and directory service:
 - Click **LDAP server** if the JNDI namespace is located on an LDAP server. You must know the following connection details:
 - The host name of the LDAP server.
 - The distinguished name of the location of the JNDI namespace.
 - Click **File system** if the JNDI namespace is located in a file system. You must know the following connection details:
 - The path to the location of the JNDI namespace in the file system.
 - Click **Other** if the JNDI namespace is located somewhere else. You must know the following connection details:
 - The name and location of the initial context factory class of the JNDI service provider.
 - The URL of the location of the JNDI namespace.
3. If the JNDI namespace is on an LDAP server, or is provided by some other JNDI service provider that requires authentication, select the type of authentication that IBM MQ Explorer must use to connect to the JNDI namespace:
 - Click **None** to use anonymous authentication to connect to the JNDI namespace. The JNDI does not pass any security credentials to the service provider from IBM MQ Explorer.
 - Click **Simple authentication** if IBM MQ Explorer must pass security credentials to the JNDI service provider.
 - Click **CRAM-MD5 authentication** if IBM MQ Explorer must pass security credentials of CRAM-MD5 standard to the JNDI service provider.
4. Optional: Edit the nickname that will be used to display the initial context in IBM MQ Explorer. By default, the location of the JNDI namespace is used but you can change it to something that is easier to read and recognize.
5. Optional: Select whether IBM MQ Explorer automatically connects to the initial context:
 - Select the **Connect immediately on finish** check box to connect to the initial context when the wizard closes.

If you clear this check box, when the wizard closes, the initial context is added to the JMS Administered Objects folder but IBM MQ Explorer cannot access the JMS objects in the JNDI namespace until you connect to the initial context.
 - Select the **Automatically reconnect to context on startup** check box if you want IBM MQ Explorer to automatically reconnect to the initial context every time that you close and re-open IBM MQ Explorer.
6. Click **Finish**.

Results

The initial context is added to the JMS Administered Objects folder in the Navigator view. If IBM MQ Explorer is connected to the initial context, you can now create connection factory objects, destination objects, and subcontexts in the initial context.

Related concepts

[“JMS contexts” on page 205](#)

A context is a set of bindings that associates names with objects stored in a naming and directory service.

Related tasks

[“Connecting and disconnecting an initial context” on page 211](#)

You can connect or disconnect IBM MQ Explorer to an initial context that is displayed in the JMS Administered Objects folder. You can also configure each initial context so that IBM MQ Explorer automatically reconnects to it the next time that you close and restart IBM MQ Explorer.

[“Removing an initial context” on page 212](#)

If you no longer want to access and administer JMS objects in a particular JNDI namespace, you can remove the initial context that defines the root of the JNDI namespace from the JMS Administered Objects folder in IBM MQ Explorer.

[“Creating a connection factory” on page 213](#)

A JMS client (a Java application that uses the JMS API) uses connection factories to create connections to the JMS provider (a messaging provider such as IBM MQ).

[“Creating a destination” on page 215](#)

A JMS client uses a destination object to specify the target of messages that the JMS client produces and the source of messages that the JMS client receives. Destination objects can represent queues (for point-to-point messaging) or topics (for publish/subscribe messaging).

[“Creating a subcontext” on page 220](#)

A subcontext is a subdivision of a JNDI namespace and can contain connection factories and destinations as well as other subcontexts. You can create subcontexts within initial contexts or within other subcontexts.

Connecting and disconnecting an initial context

You can connect or disconnect IBM MQ Explorer to an initial context that is displayed in the JMS Administered Objects folder. You can also configure each initial context so that IBM MQ Explorer automatically reconnects to it the next time that you close and restart IBM MQ Explorer.

About this task

To administer IBM MQ classes for JMS objects in IBM MQ Explorer, IBM MQ Explorer must be connected to the initial context that defines the root of the JNDI namespace in which the JMS objects are stored. If IBM MQ Explorer is disconnected from the initial context, the context is still displayed in the JMS Administered Objects folder but you cannot view or administer the objects in JNDI namespace.

- [“Connecting and disconnecting an initial context” on page 211](#)
- [“Automatically reconnecting to an initial context” on page 212](#)

Connecting and disconnecting an initial context

About this task

If the JNDI namespace is on a different computer to IBM MQ Explorer, ensure that the naming and directory service is available.

To connect or disconnect IBM MQ Explorer to an initial context that is displayed in the JMS Administered Objects folder:

Procedure

1. In the Navigator view, right-click the initial context, then click **Connect** or **Disconnect** as required.
2. If the JNDI service provider requires authentication (for example, LDAP), enter your authentication details when prompted.

Results

IBM MQ Explorer connects or disconnects the initial context. The color of the initial context's icon changes to show its status: gray if it is disconnected; blue if it is connected.

If you disconnect an initial context that is configured so that IBM MQ Explorer automatically reconnects to it, the next time that you close and restart IBM MQ Explorer, the initial context is reconnected.

What to do next

If you want to remove the initial context completely from IBM MQ Explorer, see [Removing an initial context](#).

Automatically reconnecting to an initial context

About this task

You can configure each initial context so that IBM MQ Explorer automatically reconnects to it the next time that you close and restart IBM MQ Explorer. If you do not configure an initial context to automatically reconnect, when you close and restart IBM MQ Explorer, it is not reconnected.

To configure an initial context so that IBM MQ Explorer automatically reconnects to it on startup, perform one of the following tasks:

To configure the initial context so that IBM MQ Explorer does not automatically reconnect to it:

Procedure

1. When you add an initial context to the JMS Administered Objects folder in IBM MQ Explorer, you can select the **Automatically reconnect on startup** check box in the **Add New Context** wizard.
2. For initial contexts that are already shown in the JMS Administered Objects folder, right-click the initial context, then click **Autoreconnect**. A check mark is placed next to the menu item to indicate that the initial context is set to automatically reconnect to IBM MQ Explorer on startup.
3. In the Navigator view, right-click the initial context, then click **Autoreconnect**. The check mark next to the menu item is removed.

Results

If the initial context is in a naming and directory service that requires authentication (for example, an LDAP server), when you start IBM MQ Explorer, you are prompted for the authentication details for each initial context that requires authentication and is set to automatically reconnect.

Related concepts

[“JMS contexts” on page 205](#)

A context is a set of bindings that associates names with objects stored in a naming and directory service.

Related tasks

[“Adding an initial context” on page 209](#)

To create and configure JMS objects in IBM MQ Explorer, you must add an initial context to define the root of the JNDI namespace in which the JMS objects are stored in the naming and directory service.

[“Removing an initial context” on page 212](#)

If you no longer want to access and administer JMS objects in a particular JNDI namespace, you can remove the initial context that defines the root of the JNDI namespace from the JMS Administered Objects folder in IBM MQ Explorer.

Removing an initial context

If you no longer want to access and administer JMS objects in a particular JNDI namespace, you can remove the initial context that defines the root of the JNDI namespace from the JMS Administered Objects folder in IBM MQ Explorer.

About this task

Removing the initial context from the JMS Administered Objects folder in IBM MQ Explorer does not delete the JNDI namespace and the objects that it contains from the naming and directory service. If you later want to administer the JMS objects using IBM MQ Explorer, you can add the initial context again; for more information, see [Adding an initial context](#).

If you do not want to administer JMS objects in the JNDI namespace now but intend to do so later, you can disconnect from the initial context without removing it from IBM MQ Explorer; for more information, see [Connecting and disconnecting an initial context](#).

To remove an initial context from IBM MQ Explorer:

Procedure

1. In the Navigator view, right-click the initial context, then click **Remove**
2. When prompted, click **Yes**.

Results

The initial context is removed from the JMS Administered Objects folder in IBM MQ Explorer. The JNDI namespace is not deleted from the naming and directory service so you can add the initial context to IBM MQ Explorer again later.

Related concepts

[“JMS contexts” on page 205](#)

A context is a set of bindings that associates names with objects stored in a naming and directory service.

Related tasks

[“Adding an initial context” on page 209](#)

To create and configure JMS objects in IBM MQ Explorer, you must add an initial context to define the root of the JNDI namespace in which the JMS objects are stored in the naming and directory service.

[“Connecting and disconnecting an initial context” on page 211](#)

You can connect or disconnect IBM MQ Explorer to an initial context that is displayed in the JMS Administered Objects folder. You can also configure each initial context so that IBM MQ Explorer automatically reconnects to it the next time that you close and restart IBM MQ Explorer.

Creating a connection factory

A JMS client (a Java application that uses the JMS API) uses connection factories to create connections to the JMS provider (a messaging provider such as IBM MQ).

About this task

When you define a connection factory, you select the messaging provider that is to be used as the JMS provider. If you want to change the JMS provider, you must create a new connection factory for the new JMS provider.

The initial context for the JNDI namespace in which you want to create the connection factory must be displayed in the JMS Administered Objects folder and must be connected to IBM MQ Explorer.

To create a connection factory object, complete the following steps.

Procedure

1. In the Navigator view, expand the **JMS Administered Objects** folder, then expand the initial context (and subcontexts, if necessary) for the JNDI namespace in which the connection factory will be stored.
2. Right-click the **Connection Factories** folder, then click **New > Connection Factory....** The New Connection Factory wizard opens.
3. In the wizard, type a name for the connection factory and select the messaging provider to which the JMS client will use the connection factory to connect, then click **Next**:
 - If you are using point-to-point messaging or if you are using the IBM MQ Publish/Subscribe broker, click **IBM MQ**.
4. Select the type of connection factory that you want to create:

- Click **Connection Factory** if the JMS application will use both point-to-point messaging and publish/subscribe messaging, especially if you want the JMS application to perform both types of messaging under the same transaction.
 - Click **Queue Connection Factory** if the JMS application will use only point-to-point messaging.
 - Click **Topic Connection Factory** if the JMS application will use only publish/subscribe messaging.
5. Optional: To support XA transactions, select the **Support XA transactions** check box. XA transactions are not supported if you are using Real-time as the messaging provider.
 6. Click **Next**.
 7. Select the type of transport that will be used by the connections that are created by the connection factory, then click **Next**:
 - If the JMS client that uses the connection factory is on a different computer from the queue manager, click **MQ Client**. This means that the connection uses TCP/IP. If you select **MQ Client** and you selected the Support XA transactions check box on the previous page of the wizard, you must install the Java Extended Transaction Support component of IBM MQ.
 - If the JMS application using the connection factory runs on the same computer as the queue manager, you can click **MQ Client** (see the previous option for more information) or you can click **Bindings**, which means that the JMS client connects directly to the queue manager.
 - If the JMS client that uses the connection factory will connect to the Real-time broker using TCP/IP, click **Direct**.
 - If the JMS client that uses the connection factory will connect to the Real-time broker using HTTP tunneling, click **Direct HTTP**.

Note: Real-time transport is not available in IBM MQ 8.0. If you are using IBM MQ 8.0 you can define Real-time transport, but it fails when an attempt is made to create a connection.
 8. Optional: If you want to create the connection factory object with the same attributes as an existing connection factory, select the **Create with attributes like an existing JNDI object**, then click **Select** to select the existing connection factory.
 9. Click **Finish**.

Results

The new connection factory is displayed in the Content view of the Connection Factories folder.

Related concepts

[“JMS connection factories” on page 206](#)

A connection factory is an object that a JMS client (a JMS program that uses the JMS API) uses to create a connection with a JNDI provider (a messaging provider such as IBM MQ).

Related tasks

[“Creating a destination” on page 215](#)

A JMS client uses a destination object to specify the target of messages that the JMS client produces and the source of messages that the JMS client receives. Destination objects can represent queues (for point-to-point messaging) or topics (for publish/subscribe messaging).

[“Creating a subcontext” on page 220](#)

A subcontext is a subdivision of a JNDI namespace and can contain connection factories and destinations as well as other subcontexts. You can create subcontexts within initial contexts or within other subcontexts.

[“Changing the transport type used for connections” on page 219](#)

You can change the transport type that a JMS client uses to connect to a JMS provider. You might also need to change any properties and settings that are required by the new transport type.

[“Deleting an administered object” on page 222](#)

When you delete an administered object in IBM MQ Explorer, the administered object no longer exists in the JNDI namespace in the naming and directory service.

[“Renaming an administered object” on page 220](#)

When you have created an administered object (connection factories and destinations), you can subsequently rename it in IBM MQ Explorer.

Creating a destination

A JMS client uses a destination object to specify the target of messages that the JMS client produces and the source of messages that the JMS client receives. Destination objects can represent queues (for point-to-point messaging) or topics (for publish/subscribe messaging).

Before you begin

- [Add the initial context in which you want to create the destination.](#)
- [Connect to the initial context.](#)

About this task

To create a destination object, complete the following steps.

Procedure

1. In the Navigator view, expand the **JMS Administered Objects** folder, then expand the initial context (and subcontexts, if necessary) in which the destination will be stored in the JNDI naming and directory service.
2. Right-click the **Destinations** folder, then click **New > Destination**. The New Destination wizard opens.
3. In the wizard, type a name for the destination, then click **Next**.
4. Select the type of destination that you want to create, then click **Next**:
 - If you are using point-to-point messaging, click **Queue**.
 - If you are using publish/subscribe messaging, click **Topic**.
5. Optional: If you want to create the destination object with the same attributes as an existing destination, select the **Create with attributes like an existing JNDI object**, then click **Select** to select the existing destination.
6. Click **Finish**.

Results

The new destination is displayed in the Content view of the Destinations folder.

Related concepts

[“JMS destinations \(queues and topics\)” on page 208](#)

A JMS destination is an object (a JMS queue or a JMS topic) that represents the target of messages that the client produces and the source of messages that the client consumes. In point-to-point messaging, destinations represent queues; in publish/subscribe messaging, destinations represent topics.

Related tasks

[“Creating a connection factory” on page 213](#)

A JMS client (a Java application that uses the JMS API) uses connection factories to create connections to the JMS provider (a messaging provider such as IBM MQ).

[“Creating a subcontext” on page 220](#)

A subcontext is a subdivision of a JNDI namespace and can contain connection factories and destinations as well as other subcontexts. You can create subcontexts within initial contexts or within other subcontexts.

[“Deleting an administered object” on page 222](#)

When you delete an administered object in IBM MQ Explorer, the administered object no longer exists in the JNDI namespace in the naming and directory service.

[“Renaming an administered object” on page 220](#)

When you have created an administered object (connection factories and destinations), you can subsequently rename it in IBM MQ Explorer.

[“Creating a JMS object from an IBM MQ object” on page 217](#)

You can create new JMS administered objects based on your existing IBM MQ objects.

Creating a JMS object and an IBM MQ object simultaneously

When you create a new JMS object, you can optionally create a corresponding IBM MQ object of the same type.

Before you begin

- You must have an IBM MQ queue manager. If one does not exist, you can create one as described in: [“Creating and configuring queue managers and objects” on page 27](#)
- You must have a JMS initial context. If one does not exist, you can create one as described in: [Add the initial context that will contain the JMS topic](#)
- You must be connected to the JMS initial context as described in: [Connect to the initial context](#)

About this task

With the object creation wizards in IBM MQ Explorer, you can create an IBM MQ object and a JMS object simultaneously. You begin by launching your required object wizard, for example; an IBM MQ queue. Then you select the option to launch another wizard, for example; a JMS queue, once the object has been created. The second object wizard must be of the same object type, and the properties of one are mapped to the other.

Creating a JMS queue and then creating an IBM MQ queue simultaneously

About this task

When you create a new JMS queue in IBM MQ Explorer, you can choose to launch the IBM MQ **New Local Queue** wizard to create an IBM MQ queue immediately after the **New JMS Destination** wizard has finished. The **New Local Queue** wizard now contains the details you entered when creating the JMS queue.

To create a new JMS queue and IBM MQ queue simultaneously in IBM MQ Explorer:

Procedure

1. Select the JMS Initial Context you want to add a new JMS queue to in the Navigator view, and right-click on its **Destinations** initial context object folder.
2. Click **New > Destination** to open the **New Destination** wizard.
3. Type a name for your queue, then select Queue in the **Type** field.
4. Select **Start wizard to create a matching MQ Queue**. Continue through the wizard to create your queue.

Results

Once you have completed the **New Destination** wizard, the **New MQ Queue** wizard opens, with many of the JMS queue details mapped to the IBM MQ queue.

Creating a JMS topic and then creating an IBM MQ topic simultaneously

About this task

When you create a new JMS topic in IBM MQ Explorer, you can choose to launch the IBM MQ **New Topic** wizard to create an IBM MQ topic immediately after the **New JMS Destination** wizard has finished. The **New Topic** wizard now contains the details you entered when creating the JMS topic.

To create a new JMS topic and IBM MQ topic simultaneously in IBM MQ Explorer:

Procedure

1. Select the JMS Initial Context you want to add a new JMS topic to in the Navigator view, and right-click on its **Destinations** initial context object folder.
2. Click **New > Destination** to open the **New Destination** wizard.
3. Type a name for your topic, then select Topic in the **Type** field.
4. Select **Start wizard to create a matching MQ Topic**. Continue through the wizard to create your topic.

Results

Once you have completed the **New Destination** wizard, the **New Topic** wizard opens, with many of the JMS topic details mapped to the IBM MQ topic.

Related tasks

[“Creating a destination” on page 215](#)

A JMS client uses a destination object to specify the target of messages that the JMS client produces and the source of messages that the JMS client receives. Destination objects can represent queues (for point-to-point messaging) or topics (for publish/subscribe messaging).

[“Creating and configuring queue managers and objects” on page 27](#)

You can create, configure, and delete queue managers and objects in IBM MQ Explorer by using the Navigator view and Content view.

[“Creating an IBM MQ object from a JMS object” on page 49](#)

You can create new IBM MQ queues and topics based on your existing JMS queues and topics. The values of relevant properties of the JMS object are copied to the new IBM MQ object. In future, however, if you make a change to one of the objects, the changes are not reflected in the other object.

[“Creating a JMS object from an IBM MQ object” on page 217](#)

You can create new JMS administered objects based on your existing IBM MQ objects.

Related reference

[“Destination properties” on page 590](#)

You can view and set destination properties in the Destination properties dialog. The properties that are available in the dialog depend on the type of destination.

[“Connection factory properties” on page 561](#)

You can view and set connection factory properties in the Connection Factory properties dialog. The properties that are available in the dialog depend on which messaging provider the connection factory uses.

Creating a JMS object from an IBM MQ object

You can create new JMS administered objects based on your existing IBM MQ objects.

Before you begin

- [Show the queue manager that hosts the IBM MQ queue or topic.](#)
- [Create and configure queue managers and objects.](#)
- [Add an initial context.](#)

About this task

To create a JMS administered object from an existing IBM MQ object, complete the following steps.

Procedure

1. In the Navigator view, expand the queue manager that hosts the IBM MQ object (either an IBM MQ queue or IBM MQ topic), then click the **Queues** or **Topics** folder as appropriate to list the objects in the Content view.
2. In the Content view, right-click the object, then click **Create JMS Queue** or **Create JMS Topic** as appropriate.
The New Destination wizard opens.
3. In the wizard, click **Select**, then select the JMS context in which you want to create the new JMS object.
The JMS context's name is displayed in the **JMS Context** field of the wizard.
4. Work through the wizard to define the new JMS object, then click **Finish**.

Results

The new JMS administered object is created and displayed under the appropriate JMS context in IBM MQ Explorer.

What to do next

You can now continue to configure the JMS object as necessary.

To create a JMS object and an IBM MQ object simultaneously, follow the instructions in: [“Creating a JMS object and an IBM MQ object simultaneously” on page 216](#) or [“Creating an IBM MQ object and a JMS object simultaneously” on page 50](#)

Related tasks

[“Creating a destination” on page 215](#)

A JMS client uses a destination object to specify the target of messages that the JMS client produces and the source of messages that the JMS client receives. Destination objects can represent queues (for point-to-point messaging) or topics (for publish/subscribe messaging).

[“Creating a JMS object and an IBM MQ object simultaneously” on page 216](#)

When you create a new JMS object, you can optionally create a corresponding IBM MQ object of the same type.

[“Creating an IBM MQ object and a JMS object simultaneously” on page 50](#)

When you create a new IBM MQ object, you can optionally create a corresponding JMS object of the same type.

Copying an administered object

To copy an administered object in IBM MQ Explorer, you create a new object based on the existing object that you want to copy.

About this task

To create a copy of an existing JMS administered object, complete the following steps.

Procedure

1. In the Navigator view, right-click the relevant object folder, then open the New wizard.
For example, right-click the **Connection Factories** folder, then click **New > Connection Factory**.
2. Select the options that you require until you get to page of the wizard on which you can choose to create the object with attributes like an existing object.
3. Select the **Create with attributes like an existing object** check box.
4. Click **Select** The Select the Like Object dialog opens. The dialog lists all of the objects in the JNDI namespace that match the selections you have already made in the wizard. For example, if you are

creating a connection factory, the dialog lists all the connection factories that use the same transport type, messaging provider, and connection factory class as the one you are creating.

5. Click the object that you want to base the new object on, then click **OK**.
6. Click **Finish** to create the object.

Results

The new object is created with the same attributes as the existing object that you specified.

Changing the transport type used for connections

You can change the transport type that a JMS client uses to connect to a JMS provider. You might also need to change any properties and settings that are required by the new transport type.

About this task

A JMS client (a Java application that uses the JMS API) uses a connection factory to create a connection with the JMS provider. The messaging provider that is used as the JMS provider determines which types of transport are available to use for the connection.

To change the transport used by the JMS client to connect to the JMS provider, complete the following steps.

Procedure

1. In the Navigator view, click the **Connection Factories** folder that contains the connection factory for which you want to change the transport type. The connection factory is displayed in the Content view.
2. In the Content view, right-click the connection factory, then click **Switch Transport**.
3. Click the available transport type:
 - If you are using IBM MQ as the messaging provider, click either **Bindings** or **Client**.
 - If you are using Real-time transport as the messaging provider, click either **Direct** or **Direct HTTP**. Real-time transport is not available in IBM MQ 8.0. If you are using IBM MQ 8.0 you can define Real-time transport, but it fails when an attempt is made to create a connection.
4. Check whether you need to change any properties and settings that are required by the new transport type; for example, if you are changing from **Bindings** to **Client**, you need to set up the client connection if you have not done so already.

Results

The next time that a connection factory creates a connection for the JMS client, the connection uses the new type of transport.

Related concepts

[“Messaging providers for IBM MQ classes for JMS” on page 209](#)

A JMS client (a Java application that uses the JMS API) uses a connection factory to create a connection with the JMS provider. The messaging provider that is used as the JMS provider determines which types of transport are available to use for the connection.

[“JMS connection factories” on page 206](#)

A connection factory is an object that a JMS client (a JMS program that uses the JMS API) uses to create a connection with a JNDI provider (a messaging provider such as IBM MQ).

Related tasks

[“Creating a connection factory” on page 213](#)

A JMS client (a Java application that uses the JMS API) uses connection factories to create connections to the JMS provider (a messaging provider such as IBM MQ).

Creating a subcontext

A subcontext is a subdivision of a JNDI namespace and can contain connection factories and destinations as well as other subcontexts. You can create subcontexts within initial contexts or within other subcontexts.

About this task

A subcontext extends the initial context's naming convention. The extended naming convention is used to organize the administered objects in the JNDI namespace.

To create a new subcontext within an initial context or within another subcontext, complete the following steps.

Procedure

1. In the Navigator view, right-click the initial context or subcontext in which you want to create the new subcontext, then click **New > Subcontext...** The New Subcontext dialog opens.
2. Type a name for the new subcontext, then click **OK**.

Results

The new subcontext is displayed in the Navigator view, under the initial context or subcontext in which you created it.

Related concepts

[“JMS contexts” on page 205](#)

A context is a set of bindings that associates names with objects stored in a naming and directory service.

Related tasks

[“Deleting a subcontext” on page 222](#)

When you delete a subcontext in IBM MQ Explorer, the subcontext no longer exists in the JNDI namespace.

[“Adding an initial context” on page 209](#)

To create and configure JMS objects in IBM MQ Explorer, you must add an initial context to define the root of the JNDI namespace in which the JMS objects are stored in the naming and directory service.

Renaming an administered object

When you have created an administered object (connection factories and destinations), you can subsequently rename it in IBM MQ Explorer.

About this task

To rename an administered object, complete the following steps.

Procedure

1. In the Content view, right-click the object that you want to rename, then click **Rename** The Rename Object dialog opens.
2. Type a new name for the object, then click **OK**.

Results

The renamed object is displayed in the Content view.

Related concepts

[“JMS connection factories” on page 206](#)

A connection factory is an object that a JMS client (a JMS program that uses the JMS API) uses to create a connection with a JNDI provider (a messaging provider such as IBM MQ).

[“JMS destinations \(queues and topics\)” on page 208](#)

A JMS destination is an object (a JMS queue or a JMS topic) that represents the target of messages that the client produces and the source of messages that the client consumes. In point-to-point messaging, destinations represent queues; in publish/subscribe messaging, destinations represent topics.

[“JMS contexts” on page 205](#)

A context is a set of bindings that associates names with objects stored in a naming and directory service.

Related tasks

[“Renaming a context” on page 221](#)

You can rename a subcontext, provided that you have first deleted from the subcontext any objects that are stored in the subcontext.

Renaming a context

You can rename a subcontext, provided that you have first deleted from the subcontext any objects that are stored in the subcontext.

About this task

After you have added an initial context to IBM MQ Explorer, you cannot change its nickname; you must remove then add the initial context to IBM MQ Explorer with the new nickname. You can, however, change the name of a subcontext.

To rename a subcontext, complete the following steps.

Procedure

1. Delete from the subcontext any objects that are stored in the subcontext, including IBM MQ classes for JMS connection factories and destinations, other subcontexts, and any other objects that are shown in the Content view of the initial context.
2. Refresh the Content view of the subcontext so that IBM MQ Explorer has up-to-date information about the content of the JNDI namespace.
3. In the Navigator view, right-click the subcontext, then click **Rename**
The Rename dialog opens. If the **Rename** menu item is not available, there are still objects in the subcontext; the objects might not be displayed in IBM MQ Explorer; refresh the Content view to ensure that IBM MQ Explorer has up-to-date information about the content of the JNDI namespace.
4. Type the new name for the subcontext, then click **OK**.

Results

The subcontext is renamed.

Related concepts

[“JMS connection factories” on page 206](#)

A connection factory is an object that a JMS client (a JMS program that uses the JMS API) uses to create a connection with a JNDI provider (a messaging provider such as IBM MQ).

[“JMS destinations \(queues and topics\)” on page 208](#)

A JMS destination is an object (a JMS queue or a JMS topic) that represents the target of messages that the client produces and the source of messages that the client consumes. In point-to-point messaging, destinations represent queues; in publish/subscribe messaging, destinations represent topics.

[“JMS contexts” on page 205](#)

A context is a set of bindings that associates names with objects stored in a naming and directory service.

Related tasks

[“Renaming an administered object” on page 220](#)

When you have created an administered object (connection factories and destinations), you can subsequently rename it in IBM MQ Explorer.

Deleting an administered object

When you delete an administered object in IBM MQ Explorer, the administered object no longer exists in the JNDI namespace in the naming and directory service.

About this task

Important: Before you delete an administered object, make sure that none of your JMS client applications need it. Any JMS client applications that do still need the administered object after you delete it will no longer work properly.

To delete an administered object, complete the following steps.

Procedure

1. In the Content view, right-click the administered object, then click **Delete**
2. When you are prompted, click **Delete** to confirm that you want to delete the administered object.

Results

The administered object is deleted from the JNDI namespace as well as from IBM MQ Explorer.

Related concepts

[“JMS connection factories” on page 206](#)

A connection factory is an object that a JMS client (a JMS program that uses the JMS API) uses to create a connection with a JNDI provider (a messaging provider such as IBM MQ).

[“JMS destinations \(queues and topics\)” on page 208](#)

A JMS destination is an object (a JMS queue or a JMS topic) that represents the target of messages that the client produces and the source of messages that the client consumes. In point-to-point messaging, destinations represent queues; in publish/subscribe messaging, destinations represent topics.

[“JMS contexts” on page 205](#)

A context is a set of bindings that associates names with objects stored in a naming and directory service.

Deleting a subcontext

When you delete a subcontext in IBM MQ Explorer, the subcontext no longer exists in the JNDI namespace.

About this task

You cannot delete a subcontext that contains administered objects; you must first delete from the subcontext all of the connection factories, destinations, and subcontexts in the subcontext.

Important: Before you delete a subcontext, make sure that none of your JMS client applications need any of the administered objects in subcontext. After you delete the subcontext, any JMS client applications that do still need these administered objects that were previously in the subcontext will no longer work properly.

To delete a subcontext, complete the following steps.

Procedure

1. Delete from the subcontext any objects that are stored in the subcontext, including IBM MQ classes for JMS connection factories and destinations, other subcontexts, and any other objects that are shown in the Content view of the initial context.
2. Refresh the Content view of the subcontext so that IBM MQ Explorer has up-to-date information about the content of the JNDI namespace.
3. In the Navigator view, right-click the subcontext, then click **Delete...**
If the **Delete...** menu item is not available, there are still objects in the subcontext; the objects might not be displayed in IBM MQ Explorer; refresh the Content view to ensure that IBM MQ Explorer has up-to-date information about the content of the JNDI namespace.
4. When you are prompted, click **Delete** to confirm that you want to delete the subcontext.

Results

The subcontext is deleted from the JNDI namespace as well as from IBM MQ Explorer.

Related concepts

[“JMS contexts” on page 205](#)

A context is a set of bindings that associates names with objects stored in a naming and directory service.

[“JMS connection factories” on page 206](#)

A connection factory is an object that a JMS client (a JMS program that uses the JMS API) uses to create a connection with a JNDI provider (a messaging provider such as IBM MQ).

[“JMS destinations \(queues and topics\)” on page 208](#)

A JMS destination is an object (a JMS queue or a JMS topic) that represents the target of messages that the client produces and the source of messages that the client consumes. In point-to-point messaging, destinations represent queues; in publish/subscribe messaging, destinations represent topics.

Related tasks

[“Deleting an administered object” on page 222](#)

When you delete an administered object in IBM MQ Explorer, the administered object no longer exists in the JNDI namespace in the naming and directory service.

Configuring IBM MQ Explorer

Use this information to help you to configure your IBM MQ Explorer installation.

About this task

You can customize the way that IBM MQ Explorer presents information to you, including the order of columns in tables and the objects that are displayed in the **Content** view. You can also customize a range of other preferences to suit your business needs. The availability of certain preferences depends on which plugins you have that are installed and enabled for IBM MQ Explorer.

Key settings can be configured in the **Preferences** dialog. To open the **Preferences** dialog: Click **Window** > **Preferences**. Other settings can be configured as per the information in the following table:

Table 4. Configuring IBM MQ Explorer preferences		
To configure	Task	See
Authorization service	Displaying object authority settings as text	“Displaying object authority settings as text” on page 260

Table 4. Configuring IBM MQ Explorer preferences (continued)

To configure	Task	See
Client connections	Remote queue managers; specifying default values that are used to connect to remote queue managers	“Specifying the default values used to connect to remote queue managers” on page 256
	TLS Key Repositories; specifying the default location and default password of TLS certificates	“Specifying the default location and default password of TLS certificates” on page 105
	TLS Options; specifying default security preferences	“Default security preferences” on page 192
	Security exit; configuring a default security exit	“Configuring a default security exit” on page 190
	User identification; enabling default user identification	“Users and groups (entities) in the authorization service” on page 179
Display settings	Changing the colors	“Changing the colors” on page 254
	Defining schemes and filters from the relevant content view	Filtering the objects displayed in the Content view
	Setting the order of columns in tables and the objects that are displayed	Changing the order of columns in tables
	Changing the refresh frequency of queue manager information	“Changing the refresh frequency of queue manager information” on page 255
	Displaying object authority settings as text	“Displaying object authority settings as text” on page 260
Enable Plug-ins	Enabling installed plug-ins	“Enabling installed plug-ins” on page 255
Managed File Transfer	Configuring managed file transfer	“Configuring Managed File Transfer preferences” on page 330
Messages	Configuring messages	“Configuring message preferences” on page 332
Passwords	Setting password preferences	“Passwords preferences” on page 194
Telemetry	Configuring telemetry channels	“Telemetry channels” on page 270
Tests	Including hidden queue managers in test configurations	“Including hidden queue managers in test configurations” on page 259
	Including SYSTEM objects when you run tests	“Including SYSTEM objects when you run tests” on page 259

You can export and import the customizations that you make in IBM MQ Explorer. For more information, see [Exporting and importing settings in MQ Explorer](#).

Related tasks

[“Configuring IBM MQ using IBM MQ Explorer” on page 26](#)

In the Navigator view, you can use the Properties dialog to configure certain IBM MQ properties that apply to the whole installation. If necessary, you can also configure the properties of individual queue managers.

Related reference

[“Accessibility in IBM MQ Explorer” on page 309](#)

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully.

Filtering the objects displayed in tables

When object data is displayed in IBM MQ Explorer in tables, you can filter the data so that only the objects in which you are interested are displayed.

About this task

An example of the filtering that you can do is that when the Content view displays the contents of a queue manager's **Queues** folder, you might choose to display only the queues that have a name that starts with saturn. Instead of creating a new filter for each type of object on each queue manager, you can create a filter for a type of object, such as queues, on one queue manager, and then reuse the filter for queues on other queue managers.

You can filter on the following criteria:

- The name of the object.
- The type of the object (queues and channels only).
- An attribute of the object (optional).

IBM MQ Explorer supplies and applies a standard filter for each type of object. For example, the **Standard for Queues** filter includes all the queues so that you can be sure that you are viewing all the queues on the queue manager, the **Standard for Channels** filter includes all the channels, and so on. IBM MQ also supplies a selection of other useful filters; for example, an **All queues with messages** filter displays only queues that have one or more messages.

To apply a different filter to a folder of objects:

Procedure

1. In the Content view or dialog that contains the table, click the small arrow next to the current filter name. A menu is displayed.
2. If you want to apply one of the other supplied filters, in the menu, click the name of the filter. The menu closes and the filter is applied to the table.
3. If you want to apply a different filter (that was not supplied with IBM MQ), click **More Filters...** The Select Filter dialog opens displaying the filters that are available.
4. In the **Apply filter** list, click the filter that you want to apply, or click **No filter** to remove all filtering from the table.
5. Click **OK**.

Results

The selected filter is applied to the selected folder.

Related concepts

[“Define schemes to change the order of columns in tables” on page 250](#)

When object data is displayed in IBM MQ Explorer in tables, you can customized the order of the columns in the tables.

Creating a filter

About this task

You can create filters for any IBM MQ object, including channels, queues, listeners, and services. A filter can be applied only to one type of object because different types of objects have different attributes.

The following instructions use an example of creating a filter for queues so that only local queues with names that start with `jupiter` and that contain more than 50 messages are displayed in the **Content** view. You can easily adapt the instructions to create filters for other types of object too.

To create a new filter:

Procedure

1. In the **Content** view or dialog that contains the table, click the small arrow next to the current filter name. A menu is displayed.
2. From the menu, click **Manage Filters**. The **Manage Filters** dialog opens displaying the filters that already exist for the object type.
3. In the **Manage Filters** dialog, click **Add**. The **Add Filter** dialog opens.
4. In the **Add Filter** dialog, in the **Filter Name** field, type a name for the filter; for example, Queues containing more than 50 messages
5. Following the **Filter Name** field, are the **Includes objects where** fields, in which you can enter the criteria to add to the new filter. For example, if you are creating a filter for queues, the fields are labeled **Includes Queues where**. Enter the following information:
 - a) The first row of fields allows you to filter on the name of the object. By default, the third field contains an asterisk (*) so that all objects, regardless of their names, are included in the filter. For example, to include only the queues that start with `jupiter`, type `jupiter*`
 - b) Queues and channels only: The next row of fields allows you to filter on the type of the object. By default, the filter includes all types of the object. For example, to include only local queues, select **Local Queue**.
 - c) Optional: You can enter another criteria to the filter based on the value of an attribute of the objects. Select the check box labeled **- and -** so that you can edit the fields. For example, to include only queues that contain more than 50 messages, in the first field, select the attribute **Current queue depth**; in the second field, select **Greater than**; in the third field, type 50.
6. Optional: To automatically apply an existing column scheme when the filter is applied, select the check box labeled **Automatically apply a Column Scheme when this filter is applied**, then select the column scheme from the list.
7. Click **OK**. The **Add Filter** dialog closes. The new filter is displayed in the **Manage Filters** dialog with any other available filters.
8. In the **Manage Filters** dialog, click **OK**. The **Manage Filters** dialog closes.

Results

You can now apply the filter to the table.

Related concepts

[“Define schemes to change the order of columns in tables” on page 250](#)

When object data is displayed in IBM MQ Explorer in tables, you can customized the order of the columns in the tables.

Related tasks

[“Filtering the objects displayed in tables” on page 225](#)

When object data is displayed in IBM MQ Explorer in tables, you can filter the data so that only the objects in which you are interested are displayed.

[“Editing an existing filter” on page 227](#)

[“Copying an existing filter” on page 228](#)

Editing an existing filter

About this task

You can edit any filters that you created previously and you can also edit the filters that are supplied with IBM MQ Explorer; for example, the `Default for Queues` filter.

- [“Editing the current filter” on page 227](#)
- [“Editing another filter” on page 227](#)

Editing the current filter

About this task

To edit the filter that is applied:

Procedure

1. In the Content view or dialog that contains the table, click the small arrow next to the current filter name.
A menu is displayed.
2. From the menu, click **Edit Current Filter**.
The Edit Filter dialog opens.
3. In the Edit Filter dialog, make your changes, then click **OK**. For more information about the fields in the dialog, see: [“Creating a filter” on page 226](#).

Results

The changes to the filter are automatically applied to any tables that are using that filter.

Editing another filter

About this task

To edit an existing filter that is not currently applied:

Procedure

1. In the Content view or dialog that contains the table, click the small arrow next to the current filter name.
A menu is displayed.
2. From the menu, click **Manage Filters**.
The Manage Filters dialog opens displaying the filters that exist for the object type.
3. In the Manager Filters dialog, click the filter that you want to edit, then click **Edit**.
The Edit Filter dialog opens.
4. In the Edit Filter dialog, add, remove, or change the criteria that are set for the filter, then click **OK**. For more information about the fields in the dialog, see: [“Creating a filter” on page 226](#).
5. Click **OK** to close the Manage Filters dialog.

Results

The changes to the filter are automatically applied to any tables that are using that filter.

Related tasks

[“Filtering the objects displayed in tables” on page 225](#)

When object data is displayed in IBM MQ Explorer in tables, you can filter the data so that only the objects in which you are interested are displayed.

[“Creating a filter” on page 226](#)

[“Copying an existing filter” on page 228](#)

Copying an existing filter

About this task

If there exists a filter that is similar to a filter that you want to create, you can copy the existing filter and then edit it as required. You can copy any filter that you created previously and you can also copy the filters that are supplied with IBM MQ Explorer; for example, the `Default for Queues` filter. You cannot copy a filter for one type of object to use on another type of object; for example, you cannot copy a filter for channels to use for filtering queues.

To copy an existing filter:

Procedure

1. Make sure that the type of object for which you are creating a filter is displayed in the Content view, then click the small arrow next to the current filter name. A menu is displayed.
2. In the Select Filter dialog, click **Manage Filters...** The Manage Filters dialog opens displaying the filters that exist for the object type.
3. In the Manage Filters dialog, click the filter that you want to copy, then click **Copy As...** The Copy Filter dialog opens.
4. In the Copy Filter dialog, type a name for the new filter, then click **OK**.
5. In the Manage Filters dialog, click **Edit...** The Edit Filter dialog opens.
6. In the Edit Filter dialog, add, remove, or change the criteria that are set for the filter, then click **OK**. For more information about the fields in the dialog, see [Creating a filter](#).
7. Click **OK** to close the Manage Filters dialog.

Results

The new filter is available to apply in the Select Filter dialog.

Related tasks

[“Filtering the objects displayed in tables” on page 225](#)

When object data is displayed in IBM MQ Explorer in tables, you can filter the data so that only the objects in which you are interested are displayed.

[“Creating a filter” on page 226](#)

[“Editing an existing filter” on page 227](#)

[“Copying an existing filter” on page 228](#)

Creating and configuring a service definition

The IBM MQ Service Definition specification provides a standard for documenting IBM MQ applications as services, using WSDL and URIs.

Service definitions simplify the reuse of IBM MQ applications in service oriented architectures. By describing applications as services, using the same formats as traditional web services, they can be managed in the same way which promotes reuse and enabling integration with standard service tooling. The service definition wizard validates required information before producing a correctly formatted

service definition, meaning you will not need to have detailed knowledge of both WSDL and the IBM MQ service definition specification which would be needed to produce service definitions manually.

Service definitions aid the cataloging and governing of IBM MQ applications by allowing the resources used by an application, such as queues and queue managers, to be easily queried and for services to be looked up dynamically at runtime. This is particularly true for customers who have unmanaged IBM MQ applications (not hosted by CICS® or an Application Server, running stand-alone on Windows, UNIX, System i®, or as batch mainframe applications) which have been developed over time with inconsistent documentation and no reliable inventory of applications.

The following topics describe how to create service definitions in IBM MQ Explorer:

- [“Adding a service definition repository” on page 229](#)
- [“Deleting a service definition repository” on page 230](#)
- [“Creating a new service definition” on page 230](#)

The following topics describe how to configure existing service definitions in IBM MQ Explorer:

- [“Deleting service definitions” on page 231](#)
- [“Viewing a service definition WSDL file” on page 232](#)
- [“Exporting a WSDL file” on page 232](#)

The following topics describe the various attributes of the properties pages for service definitions sets in IBM MQ Explorer:

- [“IBM MQ service definition properties” on page 469](#)

Note: The IBM MQ Explorer Service Definition Wizard, which was introduced in IBM WebSphere MQ 7.0, is deprecated for IBM MQ 8.0.

Adding a service definition repository

Use this information to create a new service definition repository.

Before you begin

New service definitions must be created within a file based repository. Multiple repositories can be created, each containing none, one, or more service definitions. By creating multiple repositories it is possible to group service definitions together. Repositories cannot be nested, they can only be created directly beneath **Service Definition Repositories** in the **Navigator** view.

About this task

To add a new service definition repository in the **Navigator** view:

Procedure

1. Right-click on **Service Definition Repositories** to open the menu, then click **Add Repository** to open the **Add New Service Definition Repository** dialog.
2. Type a name for the new repository and click **Finish** to close the dialog and create the new repository.

Related tasks

[“Deleting a service definition repository” on page 230](#)

Deleting a service definition repository also deletes any service definitions contained within it.

[“Creating a new service definition” on page 230](#)

The service definition wizard simplifies the process of creating service definitions and is integrated into the IBM MQ Explorer. The service definition wizard is deprecated in IBM MQ 8.0

[“Deleting service definitions” on page 231](#)

Deleting a service definition is permanent. When you delete a service definition, the service definition cannot be recovered.

Deleting a service definition repository

Deleting a service definition repository also deletes any service definitions contained within it.

Before you begin

When you delete a service definition repository, you also delete all the service definitions contained within that repository. Neither the repository or service definitions can be recovered.

If a service definition is still required, you have two options available:

- You can create a new service definition in a different repository, and select the **Create with attributes like an existing Service Definition** option in the **New Service Definition** wizard and select your required service definition.
- You can export a service definition to a new location.

About this task

To delete a repository in the **Navigator** view:

Procedure

1. Right-click on the repository you want to delete to open the menu, then click **Remove**.
A confirmation dialog opens.
2. Click **Delete** to permanently delete the repository and all its stored service definitions.
The confirmation dialog closes and the repository is deleted. It might take a few seconds for the change to be updated in the **Navigator** view.

Related tasks

[“Adding a service definition repository” on page 229](#)

Use this information to create a new service definition repository.

[“Creating a new service definition” on page 230](#)

The service definition wizard simplifies the process of creating service definitions and is integrated into the IBM MQ Explorer. The service definition wizard is deprecated in IBM MQ 8.0

[“Deleting service definitions” on page 231](#)

Deleting a service definition is permanent. When you delete a service definition, the service definition cannot be recovered.

Creating a new service definition

The service definition wizard simplifies the process of creating service definitions and is integrated into the IBM MQ Explorer. The service definition wizard is deprecated in IBM MQ 8.0

Before you begin

Service definitions simplify the reuse of IBM MQ applications in service oriented architectures. By describing applications as services, using the same formats as traditional web services, they can be managed in the same way which promotes reuse and enabling integration with standard service tooling. The service definition wizard validates required information before producing a correctly formatted service definition, meaning you will not need to have detailed knowledge of both WSDL and the IBM MQ service definition specification which would be needed to produce service definitions manually.

Service definitions aid the cataloging and governing of IBM MQ applications by allowing the resources used by an application, such as queues and queue managers, to be easily queried and for services to be looked up dynamically at runtime. This is particularly true for customers who have unmanaged IBM MQ applications (not hosted by CICS or an Application Server, running stand-alone on Windows, UNIX,

System i, or as batch mainframe applications) which have been developed over time with inconsistent documentation and no reliable inventory of applications.

You must create a service definition repository before you can define a new service definition. For more information see [“Adding a service definition repository” on page 229](#)

About this task

When you have created a repository in the **Navigator** view:

Procedure

1. Right-click the repository that you want to define a new service definition in to open the menu.
2. Click **New > New Service Definition** to open the **New Service Definition** wizard. As you work through the wizard, you can press F1 for context sensitive help (Ctrl + F1 on Linux installations)

Results

A new service definition is created inside the selected repository. You can create more than one service definition inside each repository.

What to do next

Service definition names must be unique within each repository, but can be reused in other repositories.

Related tasks

[“Deleting service definitions” on page 231](#)

Deleting a service definition is permanent. When you delete a service definition, the service definition cannot be recovered.

[“Adding a service definition repository” on page 229](#)

Use this information to create a new service definition repository.

[“Deleting a service definition repository” on page 230](#)

Deleting a service definition repository also deletes any service definitions contained within it.

Deleting service definitions

Deleting a service definition is permanent. When you delete a service definition, the service definition cannot be recovered.

About this task

To permanently delete a service definition in the **Navigator** view, complete the following steps.

Procedure

1. Select the service definition repository which holds the service definition that you want to delete.
2. In the **Content** view, right-click on the service definition that you want to delete to open the context menu, then click **Delete**.
A confirmation dialog opens.
3. Click **Delete** to permanently delete the service definition.
The confirmation dialog closes and the service definition is deleted. It might take a few seconds for the change to be updated in the **Content** view.

Related tasks

[“Creating a new service definition” on page 230](#)

The service definition wizard simplifies the process of creating service definitions and is integrated into the IBM MQ Explorer. The service definition wizard is deprecated in IBM MQ 8.0

[“Adding a service definition repository” on page 229](#)

Use this information to create a new service definition repository.

[“Deleting a service definition repository” on page 230](#)

Deleting a service definition repository also deletes any service definitions contained within it.

Viewing a service definition WSDL file

After you have created a new service definition, you can view the WSDL file that is created.

About this task

To view the contents of the WSDL service definition file, complete the following steps.

Procedure

1. In the **Navigator** view, select the service definition repository which holds the service definition that you want to view
2. In the **Content** view, right-click the service definition that you want to view to open the menu, then click **View**.

By default, the WSDL service definition file opens in a new view next to the **Navigator** view.

Related tasks

[“Creating a new service definition” on page 230](#)

The service definition wizard simplifies the process of creating service definitions and is integrated into the IBM MQ Explorer. The service definition wizard is deprecated in IBM MQ 8.0

[“Deleting service definitions” on page 231](#)

Deleting a service definition is permanent. When you delete a service definition, the service definition cannot be recovered.

Related reference

[“IBM MQ service definition properties” on page 469](#)

You can set properties and attributes for service definitions while creating a new service definition, or when editing an existing service definition.

Exporting a WSDL file

After you have created a new service definition, you can export the WSDL file to a new location.

About this task

To export the contents of the WSDL service definition file:

Procedure

1. In the **Navigator** view, select the service definition repository which holds the service definition that you want to export.
2. In the **Content** view, right-click the service definition that you want to export to open the menu, then click **Export**.

A dialog opens to specify the name and location of the exported file.

Related tasks

[“Creating a new service definition” on page 230](#)

The service definition wizard simplifies the process of creating service definitions and is integrated into the IBM MQ Explorer. The service definition wizard is deprecated in IBM MQ 8.0

[“Deleting service definitions” on page 231](#)

Deleting a service definition is permanent. When you delete a service definition, the service definition cannot be recovered.

Related reference

[“IBM MQ service definition properties” on page 469](#)

You can set properties and attributes for service definitions while creating a new service definition, or when editing an existing service definition.

Creating and configuring a queue manager set

Queue manager sets enable you to group queue managers in folders, and enable you to make actions to all the queue managers in the set. This enables you to subdivide your queue managers, for example into 'test' and 'production' sets, or into sets based on the operating system of the platform.

About this task

Grouping queue managers is particularly useful if you administer large numbers of queue managers using IBM MQ Explorer, because queue managers belonging to particular applications, departments, or companies can be grouped together.

The following topics describe how to create queue manager sets in IBM MQ Explorer:

- [“Displaying queue manager sets” on page 233](#)
- [“Defining manual sets” on page 234](#)
- [“Defining automatic sets” on page 235](#)

The following topics describe how to configure existing queue manager sets in IBM MQ Explorer:

- [“Managing filters for automatic sets” on page 236](#)
- [“Adding and removing queue managers manually” on page 239](#)
- [“Adding and removing queue managers automatically” on page 240](#)
- [“Copying an existing set” on page 245](#)
- [“Deleting a set” on page 246](#)
- [“Copying queue managers to a set” on page 247](#)
- [“Dragging queue managers” on page 248](#)
- [“Importing and exporting queue manager sets” on page 249](#)

The following topics describe the various attributes of the properties pages for queue manager sets in IBM MQ Explorer:

- [“Editing the properties of an automatic set” on page 243](#)
- [“Properties of manual sets” on page 242](#)
- [“Properties of automatic sets” on page 241](#)
- [“Editing the properties of a manual set” on page 243](#)
- [“Manage Sets properties” on page 244](#)

Displaying queue manager sets

Before you can work with queue manager sets, you must first display the sets in IBM MQ Explorer. Although the queue manager sets still exist when the sets are hidden, you are unable to manage them.

About this task

From the **Navigator** view:

1. In the Navigator view, right-click the **Queue Managers** folder to open the menu, then click **Sets > Show Sets**.

The **Show Sets** command displays a default set called **All** which cannot be edited, and always contains all the queue managers.

If you want to hide all the sets (while maintaining their definitions and groupings), to simplify the Navigator view for example:

1. In the Navigator view, right-click the **Queue Managers** folder to open the menu, then click **Sets > Hide Sets**.

The **Hide Sets** command removes all the defined sets, including the **All** set, from the Navigator view (while maintaining their definitions and groupings).

Related tasks

[“Creating and configuring a queue manager set” on page 233](#)

Queue manager sets enable you to group queue managers in folders, and enable you to make actions to all the queue managers in the set. This enables you to subdivide your queue managers, for example into 'test' and 'production' sets, or into sets based on the operating system of the platform.

[“Defining manual sets” on page 234](#)

You can define manual queue manager sets that do not contain any queue managers, and add queue managers when required.

[“Defining automatic sets” on page 235](#)

You can define queue manager sets that automatically include relevant queue managers.

[“Managing filters for automatic sets” on page 236](#)

You can add, edit, copy and delete filters to configure automatic queue manager sets.

[“Adding and removing queue managers manually” on page 239](#)

When you have created a manual queue manager set, you can manually add and remove queue managers.

[“Adding and removing queue managers automatically” on page 240](#)

You can define filters to automatically manage the membership of your queue manager sets.

[“Copying an existing set” on page 245](#)

Copy an existing set to create a new queue manager set with the same configuration.

Defining manual sets

You can define manual queue manager sets that do not contain any queue managers, and add queue managers when required.

About this task

To define a new set:

Procedure

1. In the Navigator view, right-click the **Queue Managers** folder, then click **Sets > New Set...** The **New Set** wizard opens.
2. Type a valid name for your new queue manager set. The name of the set is not constrained by the normal MQ object naming rules. The name must, however, be different from the names of any existing set.
3. Click **Manual** to add queue managers manually.
4. Select one of the following options:
 - Click **Finish** to create an empty set, or
 - Click **Next** to add queue managers to the new set.
5. On the manual selection pane, select the check box next to the corresponding queue manager name to add the queue manager to your new set. You can add multiple queue managers.
6. Click **Finish** to create your set and close the wizard.

Results

The new manual queue manager set is displayed in the Navigator view.

What to do next

If you created an empty set in step 3, you can manually add queue managers as described here: [“Adding and removing queue managers manually” on page 239](#)

Related tasks

[“Creating and configuring a queue manager set” on page 233](#)

Queue manager sets enable you to group queue managers in folders, and enable you to make actions to all the queue managers in the set. This enables you to subdivide your queue managers, for example into 'test' and 'production' sets, or into sets based on the operating system of the platform.

[“Displaying queue manager sets” on page 233](#)

Before you can work with queue manager sets, you must first display the sets in IBM MQ Explorer. Although the queue manager sets still exist when the sets are hidden, you are unable to manage them.

[“Defining manual sets” on page 234](#)

You can define manual queue manager sets that do not contain any queue managers, and add queue managers when required.

[“Defining automatic sets” on page 235](#)

You can define queue manager sets that automatically include relevant queue managers.

[“Managing filters for automatic sets” on page 236](#)

You can add, edit, copy and delete filters to configure automatic queue manager sets.

[“Adding and removing queue managers manually” on page 239](#)

When you have created a manual queue manager set, you can manually add and remove queue managers.

[“Adding and removing queue managers automatically” on page 240](#)

You can define filters to automatically manage the membership of your queue manager sets.

Defining automatic sets

You can define queue manager sets that automatically include relevant queue managers.

About this task

To define a new set:

Procedure

1. In the Navigator view, right-click the **Queue Managers** folder, then click **Sets > New Set...** The **New Set** wizard opens.
2. Type a valid name for your new queue manager set. The name of the set is not constrained by the normal MQ object naming rules. The name must, however, be different from the names of any existing set.
3. Click **Automatic** to add queue managers using automatic filters and click **Next**.
4. Select the filter you want to use from the **Available Filters** pane, and click **Add->**. The filter will be removed from the **Available Filters** pane and placed in the **Selected Filters** pane. To select multiple filters, for example Platform = Unix and Command level = 500, use one of the following options:
 - Select **matches ALL the selected filters** to add an AND statement to the filter, for example Platform = Unix -AND- Command level = 500. The wizard will not allow you to continue if you have selected conflicting filters, for example Platform = Unix -AND- Platform = Windows is not allowed.
 - Select **matches ANY of the selected filters** to add an OR statement to the filter, for example Platform = Unix -OR- Command level = 500

If you need to create, copy, edit, or delete filters, you can do so now by clicking on **Manage Filters...** and following the instructions as described in: [“Managing filters for automatic sets” on page 236](#). Alternatively, filters can be managed at a later time if required.

5. Click **Finish** to create your set and close the wizard.

Results

The new automatic queue manager set is displayed in the Navigator view.

What to do next

You can create new filters to add or remove queue managers, as well as copy, edit and delete filters as described in: [“Managing filters for automatic sets” on page 236](#)

Related tasks

[“Creating and configuring a queue manager set” on page 233](#)

Queue manager sets enable you to group queue managers in folders, and enable you to make actions to all the queue managers in the set. This enables you to subdivide your queue managers, for example into 'test' and 'production' sets, or into sets based on the operating system of the platform.

[“Displaying queue manager sets” on page 233](#)

Before you can work with queue manager sets, you must first display the sets in IBM MQ Explorer. Although the queue manager sets still exist when the sets are hidden, you are unable to manage them.

[“Defining manual sets” on page 234](#)

You can define manual queue manager sets that do not contain any queue managers, and add queue managers when required.

[“Defining automatic sets” on page 235](#)

You can define queue manager sets that automatically include relevant queue managers.

[“Managing filters for automatic sets” on page 236](#)

You can add, edit, copy and delete filters to configure automatic queue manager sets.

[“Adding and removing queue managers manually” on page 239](#)

When you have created a manual queue manager set, you can manually add and remove queue managers.

[“Adding and removing queue managers automatically” on page 240](#)

You can define filters to automatically manage the membership of your queue manager sets.

Managing filters for automatic sets

You can add, edit, copy and delete filters to configure automatic queue manager sets.

Before you begin

Before you add, edit, copy or delete filters:

1. Enable set visibility as described in: [“Displaying queue manager sets” on page 233](#).
2. Define a set for the queue managers to be added to or removed from as described in: [“Defining automatic sets” on page 235](#).

About this task

To manage filters:

Procedure

1. Right-click the set you want to manage filters for in the **Navigator** view, click **Edit Set...** to open the **Edit Set** window.
2. Click **Manage Filters** to open the **Manage Filters** window.

What to do next

The following options are available:

- [“Add a new filter” on page 237](#)

- [“Edit a filter” on page 237](#)
- [“Copy a filter” on page 238](#)
- [“Remove an existing filter” on page 238](#)

Add a new filter

About this task

You can create filters to define which queue managers are grouped in your set.

The following instructions use an example for creating a filter for queue managers.

To create a new filter:

Procedure

1. Open the **Manage Filters** dialog, as described previously in [“Managing filters for automatic sets” on page 236](#).
2. In the **Manage Filters** dialog, click **Add...** The Add Filter dialog opens.
3. In the **Add Filter** dialog, in the **Filter Name** field, type a name for the filter; for example, Queues containing more than 50 messages
4. In the **Includes queue managers where** fields, you can enter the criteria to add to the new filter. For example, enter the following information:
 - a) The first row of fields allows you to filter on the name of the queue manager. By default, the third field contains an asterisk (*) so that all queue managers, regardless of their names, are included in the filter. For example, to include only the queues that start with jupiter, type jupiter*
 - b) Optional: You can enter another criteria to the filter based on the value of an attribute of the objects. Select the check box labeled - **AND**- so that you can edit the fields. For example, to include only queue manager that have a **Description** field of Payroll, select the attribute **Payroll**; in the second field, select **equal to**; in the third field, type Payroll.
5. Optional: To automatically apply an existing column scheme when the filter is applied, select the check box labeled **Automatically apply a Column Scheme when this filter is applied**, then select the column scheme from the list.
6. Click **OK**. The Add Filter dialog closes. The new filter is displayed in the Manage Filters dialog with any other available filters.
7. In the Manage Filters dialog, click **OK**. The Manage Filters dialog closes.

Results

Your new filter will be added to the existing list of available filters.

Edit a filter

About this task

You can edit filters to define which queue managers are grouped in your set.

The following instructions use an example for editing a filter for queue managers.

To edit a filter:

Procedure

1. Open the **Manage Filters** dialog, as described previously in [“Managing filters for automatic sets” on page 236](#).
2. In the **Manage Filters** dialog, click **Edit...** The Edit Filter dialog opens.

3. In the **Edit Filter** dialog, add, remove, or change the criteria that are set for the filter, then click **OK**. For more information about the fields in the dialog, see [“Add a new filter” on page 237](#).
4. Click **OK** to close the **Manage Filters** dialog.

Results

The changes to the filter are automatically applied to any tables that are using that filter. MQ Explorer might take several seconds to apply the filters to the queue managers.

Copy a filter

About this task

If you want to create a filter that is similar to an existing one, you can copy the existing filter and then edit it as required. You can copy any filter that you have created previously and you can also copy the filters that are supplied with IBM MQ Explorer; for example, the Command `level = 500` filter.

To copy an existing filter:

Procedure

1. Open the **Manage Filters** dialog, as described previously in [“Managing filters for automatic sets” on page 236](#).
2. In the **Manage Filters** dialog, select the filter you want to copy, then click **Copy As...** The **Edit Filter** dialog opens.
3. In the **Copy Filter** dialog, type a name for the new filter, then click **OK**. The copied filter name cannot be the same as an existing filter.
4. In the **Manage Filters** dialog, click **Edit...** The **Edit Filter** dialog opens.
5. In the **Edit Filter** dialog, add, remove, or change the criteria that are set for the filter, then click **OK**. For more information about the fields in the dialog, see [“Add a new filter” on page 237](#).
6. Click **OK** to close the **Manage Filters** dialog.

Results

The new filter is available to apply.

Remove an existing filter

About this task

To delete an existing filter:

Procedure

1. Open the **Manage Filters** dialog, as described previously in [“Managing filters for automatic sets” on page 236](#).
2. In the **Manage Filters** dialog, select the filter you want to delete, then click **Remove...** The **Remove Filter** confirmation dialog opens.
3. Click **Yes** to remove the filter and close the confirmation dialog.
4. Click **OK** to close the **Manage Filters** dialog.

Results

The filter you deleted will be removed from the list of filters for the selected queue manager set. Any queue managers added to the set by the filter will no longer be a member of the set in the Navigator view.

Related links

Related tasks

[“Creating and configuring a queue manager set” on page 233](#)

Queue manager sets enable you to group queue managers in folders, and enable you to make actions to all the queue managers in the set. This enables you to subdivide your queue managers, for example into 'test' and 'production' sets, or into sets based on the operating system of the platform.

[“Displaying queue manager sets” on page 233](#)

Before you can work with queue manager sets, you must first display the sets in IBM MQ Explorer. Although the queue manager sets still exist when the sets are hidden, you are unable to manage them.

[“Defining manual sets” on page 234](#)

You can define manual queue manager sets that do not contain any queue managers, and add queue managers when required.

[“Defining automatic sets” on page 235](#)

You can define queue manager sets that automatically include relevant queue managers.

[“Adding and removing queue managers manually” on page 239](#)

When you have created a manual queue manager set, you can manually add and remove queue managers.

[“Adding and removing queue managers automatically” on page 240](#)

You can define filters to automatically manage the membership of your queue manager sets.

Adding and removing queue managers manually

When you have created a manual queue manager set, you can manually add and remove queue managers.

Before you begin

Before you start:

1. Enable set visibility as described in: [“Displaying queue manager sets” on page 233](#).
2. Define a set for the queue managers to be added to or removed from as described in: [“Defining manual sets” on page 234](#).

About this task

There are two ways to manually add and remove queue managers in the IBM MQ Explorer Navigator view.

Procedure

- To add or remove queue managers using the first method:
 - a) Right-click the set that you want to modify.
The **All** set membership cannot be modified.
 - b) Click **Set Membership...** to open the **Set Membership** dialog.
All available queue managers are displayed. Queue managers that are already a member of the set have their corresponding check box already selected.
 - c) To add the queue manager to the set, select the check box next to the corresponding queue manager name. To remove the queue manager from the set, clear the check box next to the corresponding queue manager.
You can select multiple queue managers.
 - d) Click **OK** to save your changes and close the dialog.
- To add or remove queue managers using the second method:
 - a) Right-click the queue manager you want to add to a queue manager set.
 - b) Click **Sets > Manual Set Membership...** to open the **Manual Set Membership** dialog.

- c) To add the queue manager to a queue manager set, select the check box next to the corresponding queue manager set name. To remove the queue manager from that queue manager set, clear the check box next to the corresponding queue manager set.

You can select multiple queue manager sets.

- d) Click **OK** to save your changes and close the dialog.

Results

If you added queue managers to a set or removed queue managers from the set, then the new set membership is shown in the Navigator view.

Related tasks

[“Creating and configuring a queue manager set” on page 233](#)

Queue manager sets enable you to group queue managers in folders, and enable you to make actions to all the queue managers in the set. This enables you to subdivide your queue managers, for example into 'test' and 'production' sets, or into sets based on the operating system of the platform.

[“Displaying queue manager sets” on page 233](#)

Before you can work with queue manager sets, you must first display the sets in IBM MQ Explorer. Although the queue manager sets still exist when the sets are hidden, you are unable to manage them.

[“Defining manual sets” on page 234](#)

You can define manual queue manager sets that do not contain any queue managers, and add queue managers when required.

[“Defining automatic sets” on page 235](#)

You can define queue manager sets that automatically include relevant queue managers.

[“Managing filters for automatic sets” on page 236](#)

You can add, edit, copy and delete filters to configure automatic queue manager sets.

[“Adding and removing queue managers manually” on page 239](#)

When you have created a manual queue manager set, you can manually add and remove queue managers.

[“Adding and removing queue managers automatically” on page 240](#)

You can define filters to automatically manage the membership of your queue manager sets.

Adding and removing queue managers automatically

You can define filters to automatically manage the membership of your queue manager sets.

Before you begin

Before you start:

1. Enable set visibility as described in: [“Displaying queue manager sets” on page 233](#).
2. Define a set for the queue managers to be added to or removed from as described in: [“Defining automatic sets” on page 235](#).

About this task

To add and remove queue managers:

Procedure

1. Right-click the set that you want to modify. The **All** set membership cannot be modified.
2. Click **Edit Set...** to open the **Edit Set** dialog. The current filters are displayed, allowing you to add or remove them (you can also edit, copy and delete them as described in: [“Managing filters for automatic sets” on page 236](#)).
3. Click **OK** to save your changes and close the window.

Results

If your changes to the filter added queue managers to the set or removed queue managers from the set, then the new set membership is shown in the Navigator view.

What to do next

Related tasks

[“Creating and configuring a queue manager set” on page 233](#)

Queue manager sets enable you to group queue managers in folders, and enable you to make actions to all the queue managers in the set. This enables you to subdivide your queue managers, for example into 'test' and 'production' sets, or into sets based on the operating system of the platform.

[“Displaying queue manager sets” on page 233](#)

Before you can work with queue manager sets, you must first display the sets in IBM MQ Explorer. Although the queue manager sets still exist when the sets are hidden, you are unable to manage them.

[“Defining manual sets” on page 234](#)

You can define manual queue manager sets that do not contain any queue managers, and add queue managers when required.

[“Defining automatic sets” on page 235](#)

You can define queue manager sets that automatically include relevant queue managers.

[“Managing filters for automatic sets” on page 236](#)

You can add, edit, copy and delete filters to configure automatic queue manager sets.

[“Adding and removing queue managers manually” on page 239](#)

When you have created a manual queue manager set, you can manually add and remove queue managers.

Properties of automatic sets

An automatic queue manager set has several properties that you can edit.

Descriptions of the options on the **Edit Set** dialog are listed in the following table.

Object	Description
Set Name	Enter a valid name for your queue manager set. The name of the set is not constrained by the normal IBM MQ object naming rules concerning characters, but is constrained by the IBM MQ object naming rules for length. The set name must be different from the names of any existing set.
matches ALL the selected filters	Select matches ALL the selected filters to add an AND statement to the filter, for example: Platform = Unix -AND- Command level = 500. The wizard will not allow you to continue if you have selected conflicting filters. For example, you cannot have: Platform = Unix -AND- Platform = Windows. AND statements cannot be mixed with OR statements in the filter. For example, you cannot have: Platform = Unix -OR- Platform = Windows -AND- Command level = 500
matches ANY of the selected filters	Select matches ANY of the selected filters to add an OR statement to the filter, for example: Platform = Unix -OR- Command level = 500. OR statements cannot be mixed with AND statements in the filter. For example, you cannot have: Platform = Unix -OR- Platform = Windows -AND- Command level = 500
Add->	Select the filter in the Available Filters pane that you want to add and click Add-> . The filter is removed from the Available Filters pane and placed in the Selected Filters pane.

Object	Description
<-Remove	Select the filter in the Selected Filters pane that you want to remove, and click <-Remove . The filter is removed from the Selected Filters pane and placed back in the Available Filters pane.
Manage Filters...	Click Manage Filters... to open the Manage Filters window. The process of managing filters is explained here: “Managing filters for automatic sets” on page 236 .

Related tasks

[“Editing the properties of an automatic set” on page 243](#)

You can edit the properties of an existing automatic set.

[“Creating and configuring a queue manager set” on page 233](#)

Queue manager sets enable you to group queue managers in folders, and enable you to make actions to all the queue managers in the set. This enables you to subdivide your queue managers, for example into 'test' and 'production' sets, or into sets based on the operating system of the platform.

[“Defining automatic sets” on page 235](#)

You can define queue manager sets that automatically include relevant queue managers.

[“Managing filters for automatic sets” on page 236](#)

You can add, edit, copy and delete filters to configure automatic queue manager sets.

[“Adding and removing queue managers automatically” on page 240](#)

You can define filters to automatically manage the membership of your queue manager sets.

Properties of manual sets

A manual queue manager set has only one property that you can edit.

A description of the option on the **Edit Set** dialog is listed in the following table:

Object	Description
Set Name	Enter a valid name for your queue manager set. The name of the set is not constrained by the normal IBM MQ object naming rules concerning characters, but is constrained by the IBM MQ object naming rules for length. The set name must be different from the names of any existing set.

Related tasks

[“Editing the properties of a manual set” on page 243](#)

You can edit the properties of an existing manual set.

[“Creating and configuring a queue manager set” on page 233](#)

Queue manager sets enable you to group queue managers in folders, and enable you to make actions to all the queue managers in the set. This enables you to subdivide your queue managers, for example into 'test' and 'production' sets, or into sets based on the operating system of the platform.

[“Defining manual sets” on page 234](#)

You can define manual queue manager sets that do not contain any queue managers, and add queue managers when required.

[“Adding and removing queue managers manually” on page 239](#)

When you have created a manual queue manager set, you can manually add and remove queue managers.

Editing the properties of an automatic set

You can edit the properties of an existing automatic set.

About this task

There are two ways to open the **Edit Set** dialog for automatic queue manager sets in the IBM MQ Explorer Navigator view. To open the **Edit Set** dialog using the first method:

Procedure

1. Right-click the automatic set you want to edit.
2. Click **Edit Set...** to open the **Edit Set** dialog.

Results

The **Edit Set** dialog is now open, and you can edit the properties of the automatic set.

What to do next

To open the **Edit Set** dialog using the second method:

1. Right-click **Queue Managers**
2. Click **Sets > Manage Sets** to open the **Manage Sets** dialog.
3. Select the automatic set you want to edit the properties of.
4. Click **Edit...** to open the **Edit Set** dialog for automatic sets.

The **Edit Set** dialog is now open, and you can edit the properties of the automatic set.

Related tasks

[“Creating and configuring a queue manager set” on page 233](#)

Queue manager sets enable you to group queue managers in folders, and enable you to make actions to all the queue managers in the set. This enables you to subdivide your queue managers, for example into 'test' and 'production' sets, or into sets based on the operating system of the platform.

[“Defining automatic sets” on page 235](#)

You can define queue manager sets that automatically include relevant queue managers.

[“Managing filters for automatic sets” on page 236](#)

You can add, edit, copy and delete filters to configure automatic queue manager sets.

[“Adding and removing queue managers automatically” on page 240](#)

You can define filters to automatically manage the membership of your queue manager sets.

Related reference

[“Properties of automatic sets” on page 241](#)

An automatic queue manager set has several properties that you can edit.

Editing the properties of a manual set

You can edit the properties of an existing manual set.

About this task

There are two ways to open the **Edit Set** dialog for manual queue manager sets in the IBM MQ Explorer Navigator view. To open the **Edit Set** dialog using the first method:

Procedure

1. Right-click the manual set you want to edit.

2. Click **Edit Set...** to open the **Edit Set** dialog.

Results

The **Edit Set** dialog is now open, and you can edit the properties of the manual set.

What to do next

To open the **Edit Set** dialog using the second method:

1. Right-click **Queue Managers**
2. Click **Sets > Manage Sets** to open the **Manage Sets** dialog.
3. Select the manual set you want to edit the properties of.
4. Click **Edit...** to open the **Edit Set** dialog for manual sets.

The **Edit Set** dialog is now open, and you can edit the properties of the manual set.

Related tasks

[“Creating and configuring a queue manager set” on page 233](#)

Queue manager sets enable you to group queue managers in folders, and enable you to make actions to all the queue managers in the set. This enables you to subdivide your queue managers, for example into 'test' and 'production' sets, or into sets based on the operating system of the platform.

[“Defining manual sets” on page 234](#)

You can define manual queue manager sets that do not contain any queue managers, and add queue managers when required.

[“Adding and removing queue managers manually” on page 239](#)

When you have created a manual queue manager set, you can manually add and remove queue managers.

Related reference

[“Properties of manual sets” on page 242](#)

A manual queue manager set has only one property that you can edit.

Manage Sets properties

You manage set properties through the **Manage Sets** dialog.

Right-click Queue Managers to open the **Sets** menu item, then click **Manage Sets** to open the **Manage Sets** dialog.

Descriptions of the options on the **Manage Sets** dialog are listed in the following table.

Object	Description
Add	Click Add... to open the New Set window to define a new set. Manual sets and automatic sets can be added in this way. The process for defining a new manual set is explained here: “Defining manual sets” on page 234 . The process for defining a new automatic set is explained here: “Defining automatic sets” on page 235 .
Copy As	Click Copy As... to open the Copy As dialog. Manual sets and automatic sets can be copied in this way. The process for copying defined sets is explained here: “Copying an existing set” on page 245
Edit	When a manual set is currently selected, click Edit... to open the Edit Set dialog. The name of the manual set can be changed. The name you enter must be unique, you are unable to click OK until a unique name has been entered. When an automatic set is currently selected, click Edit... to open the Edit Set window.

Object	Description
Remove	Click Remove... to remove the selected set. You will be prompted to confirm or cancel your request.

Related tasks

[“Creating and configuring a queue manager set” on page 233](#)

Queue manager sets enable you to group queue managers in folders, and enable you to make actions to all the queue managers in the set. This enables you to subdivide your queue managers, for example into 'test' and 'production' sets, or into sets based on the operating system of the platform.

[“Displaying queue manager sets” on page 233](#)

Before you can work with queue manager sets, you must first display the sets in IBM MQ Explorer. Although the queue manager sets still exist when the sets are hidden, you are unable to manage them.

[“Defining manual sets” on page 234](#)

You can define manual queue manager sets that do not contain any queue managers, and add queue managers when required.

[“Adding and removing queue managers manually” on page 239](#)

When you have created a manual queue manager set, you can manually add and remove queue managers.

[“Dragging queue managers” on page 248](#)

Queue managers can be dragged into sets, as well as dragged out of sets.

Copying an existing set

Copy an existing set to create a new queue manager set with the same configuration.

Before you begin

Before you start:

1. Enable set visibility as described in: [“Displaying queue manager sets” on page 233](#).
2. Define a set to be copied, as described in: [“Defining automatic sets” on page 235](#) or: [“Defining manual sets” on page 234](#).

About this task

To copy an existing set:

Procedure

1. In the **Navigator** view, right-click the **Queue Managers** folder, then click **Sets > Manage Sets....**
The **Manage Sets** window is opened.
2. Select the set you want to copy
3. Click **Copy As...** to open the **Copy Set** dialog.
4. Type a name for your set in the **New Set Name** field. The new set name must be unique.
5. Click **OK** to copy the set and close the dialog.

Results

You have successfully copied a set, the Navigator view is updated with the new set (This might take a few seconds if there are many queue managers in the set).

Related tasks

[“Creating and configuring a queue manager set” on page 233](#)

Queue manager sets enable you to group queue managers in folders, and enable you to make actions to all the queue managers in the set. This enables you to subdivide your queue managers, for example into 'test' and 'production' sets, or into sets based on the operating system of the platform.

[“Displaying queue manager sets” on page 233](#)

Before you can work with queue manager sets, you must first display the sets in IBM MQ Explorer. Although the queue manager sets still exist when the sets are hidden, you are unable to manage them.

[“Defining manual sets” on page 234](#)

You can define manual queue manager sets that do not contain any queue managers, and add queue managers when required.

[“Defining automatic sets” on page 235](#)

You can define queue manager sets that automatically include relevant queue managers.

[“Managing filters for automatic sets” on page 236](#)

You can add, edit, copy and delete filters to configure automatic queue manager sets.

[“Adding and removing queue managers manually” on page 239](#)

When you have created a manual queue manager set, you can manually add and remove queue managers.

[“Adding and removing queue managers automatically” on page 240](#)

You can define filters to automatically manage the membership of your queue manager sets.

Deleting a set

Deleting a queue manager set deletes the set itself, but not the queue managers within the set.

Before you begin

1. Enable set visibility as described in: [“Displaying queue manager sets” on page 233](#).
2. Define a set to be deleted, as described in: [“Defining automatic sets” on page 235](#) or in: [“Defining manual sets” on page 234](#).

About this task

When you delete a queue manager set, the queue managers within the set are NOT deleted. To delete an existing set, complete the following steps.

1. In the **Navigators** view, right-click the set you want to delete, to open the menu. Click **Delete** to open a confirmation dialog. Click **Delete** to permanently delete the selected set.

Results

You have successfully removed a set, the Navigators view is updated with the new information (This might take a few seconds if there are many queue managers in the set).

Related tasks

[“Creating and configuring a queue manager set” on page 233](#)

Queue manager sets enable you to group queue managers in folders, and enable you to make actions to all the queue managers in the set. This enables you to subdivide your queue managers, for example into 'test' and 'production' sets, or into sets based on the operating system of the platform.

[“Displaying queue manager sets” on page 233](#)

Before you can work with queue manager sets, you must first display the sets in IBM MQ Explorer. Although the queue manager sets still exist when the sets are hidden, you are unable to manage them.

[“Defining manual sets” on page 234](#)

You can define manual queue manager sets that do not contain any queue managers, and add queue managers when required.

[“Defining automatic sets” on page 235](#)

You can define queue manager sets that automatically include relevant queue managers.

[“Managing filters for automatic sets” on page 236](#)

You can add, edit, copy and delete filters to configure automatic queue manager sets.

[“Adding and removing queue managers manually” on page 239](#)

When you have created a manual queue manager set, you can manually add and remove queue managers.

[“Adding and removing queue managers automatically” on page 240](#)

You can define filters to automatically manage the membership of your queue manager sets.

Copying queue managers to a set

Copying queue managers from one set to another is an easy way to add a queue manager to a selection of sets quickly, without the need to having to add the queue manager to each set individually for example.

Before you begin

Before you start:

1. Enable set visibility as described in: [“Displaying queue manager sets” on page 233](#).
2. Define a minimum of 2 manual sets, one to be copied from and another to be copied to, as described in: [“Defining manual sets” on page 234](#). Automatic sets cannot be used with this following process.

About this task

To copy the queue managers to a set:

Procedure

1. In the **Navigator** view, right-click the set you want to copy the queue managers from, then click **Copy to set...**
The **Copy to set** dialog opens.
2. Select the check box next to the corresponding set name to add the queue managers to. You can select multiple sets.
3. Optional: You can click **Manage Sets...** to define or remove a set as is described in: [“Adding and removing queue managers manually” on page 239](#)
4. Click **OK** to close the **Copy to set** dialog.

Results

You have successfully copied the contents of one set to another. The navigator view is updated with the new information (This might take a few seconds if there are many queue managers in the set).

Related tasks

[“Creating and configuring a queue manager set” on page 233](#)

Queue manager sets enable you to group queue managers in folders, and enable you to make actions to all the queue managers in the set. This enables you to subdivide your queue managers, for example into 'test' and 'production' sets, or into sets based on the operating system of the platform.

[“Displaying queue manager sets” on page 233](#)

Before you can work with queue manager sets, you must first display the sets in IBM MQ Explorer. Although the queue manager sets still exist when the sets are hidden, you are unable to manage them.

[“Defining manual sets” on page 234](#)

You can define manual queue manager sets that do not contain any queue managers, and add queue managers when required.

[“Adding and removing queue managers manually” on page 239](#)

When you have created a manual queue manager set, you can manually add and remove queue managers.

[“Dragging queue managers” on page 248](#)

Queue managers can be dragged into sets, as well as dragged out of sets.

Dragging queue managers

Queue managers can be dragged into sets, as well as dragged out of sets.

Before you begin

Before you start:

1. Enable set visibility as described in: [“Displaying queue manager sets” on page 233](#).
2. Define a set for the queue managers to be added to or removed from, as described in: [“Defining manual sets” on page 234](#).

About this task

There are several ways to drag queue managers:

Procedure

- Drag a queue manager from the All set into a manual set to add it to that manual set. The queue manager will not be removed from the All set.
- Drag a queue manager from a manual set into the All set to remove it from the manual set.
- Drag a queue manager from a manual set into a second manual set. The queue manager will be added to the second manual set and removed from the first.
- Drag a queue manager from an automatic set into a manual set to add it to the manual set. The queue manager will not be removed from the automatic set.
- Drag a queue manager from a manual set into a second manual set while holding down the Ctrl key. The queue manager will be added to the second manual set and remain in the first.

Example

What to do next

Queue managers cannot be dragged into an automatic set from another set. Queue managers cannot be dragged from an automatic set into the All set, for example: You cannot remove a queue manager from an automatic set by dragging it.

Related tasks

[“Creating and configuring a queue manager set” on page 233](#)

Queue manager sets enable you to group queue managers in folders, and enable you to make actions to all the queue managers in the set. This enables you to subdivide your queue managers, for example into 'test' and 'production' sets, or into sets based on the operating system of the platform.

[“Displaying queue manager sets” on page 233](#)

Before you can work with queue manager sets, you must first display the sets in IBM MQ Explorer. Although the queue manager sets still exist when the sets are hidden, you are unable to manage them.

[“Defining manual sets” on page 234](#)

You can define manual queue manager sets that do not contain any queue managers, and add queue managers when required.

[“Adding and removing queue managers manually” on page 239](#)

When you have created a manual queue manager set, you can manually add and remove queue managers.

[“Copying queue managers to a set” on page 247](#)

Copying queue managers from one set to another is an easy way to add a queue manager to a selection of sets quickly, without the need to having to add the queue manager to each set individually for example.

Importing and exporting queue manager sets

About this task

You can export your queue manager sets from IBM MQ Explorer for backup purposes, or to transfer and import the queue manager sets into another instance of IBM MQ Explorer.

Exporting queue manager sets

About this task

To export queue manager sets from IBM MQ Explorer:

Procedure

1. In the Navigator view, right-click **IBM MQ**, then click **Export MQ Explorer settings...** The **Export** dialog opens.
2. Select **Sets** from the check boxes.
3. Enter the file name and location for the compressed file that is created to store the exported queue manager sets.
4. Click **OK**.

Results

A compressed file that contains the exported queue manager sets is created. The compressed file contains the settings in XML files.

When you are exporting manual queue manager sets, a list of the names of the queue managers that are members of the set, and the QMID of the queue managers, is exported.

When you are exporting automatic queue manager sets, a list of identifiers for filters that queue managers must match, and whether queue managers must match any or all of the filters, is exported.

Importing queue manager sets

About this task

To import queue manager sets into IBM MQ Explorer:

Procedure

1. In the Navigator view, right-click **IBM MQ**, then click **Import MQ Explorer settings...** The **Import** dialog opens.
2. Browse for the compressed file that contains the queue manager sets.
3. Select **Sets** to import the settings. If the compressed file does not contain any exported queue manager set information, the check box associated with sets is unavailable.
4. Click **OK**.

Results

The settings from the compressed file are imported into IBM MQ Explorer. The queue manager appends the imported sets with the currently defined sets, such that the sets are merged together in IBM MQ Explorer.

When you are importing manual queue manager sets, a list of the names of the queue managers that are members of the set, and the QMID of the queue managers, is imported. Any existing queue manager with

a QMID that matches a QMID from the imported set definition is added to the queue manager set. If IBM MQ Explorer does not contain a queue manager with a matching QMID, then the imported information for that queue manager is ignored.

When you are importing automatic queue manager sets, a list of identifiers for filters that queue managers must match, and whether queue managers must match any or all of the filters, is imported. Only existing queue managers that match the imported filter rules are added to the corresponding automatic set. If any filters are missing, a dialog is displayed asking you to either select a different filter or delete the set.

Queue manager set definitions cannot be imported into IBM WebSphere MQ MQ Explorer 6.0 or earlier.

Related tasks

[“Creating and configuring a queue manager set” on page 233](#)

Queue manager sets enable you to group queue managers in folders, and enable you to make actions to all the queue managers in the set. This enables you to subdivide your queue managers, for example into 'test' and 'production' sets, or into sets based on the operating system of the platform.

[“Displaying queue manager sets” on page 233](#)

Before you can work with queue manager sets, you must first display the sets in IBM MQ Explorer.


Although the queue manager sets still exist when the sets are hidden, you are unable to manage them.

Define schemes to change the order of columns in tables

When object data is displayed in IBM MQ Explorer in tables, you can customized the order of the columns in the tables.

For example, queues have a large number of attributes; when you display the contents of a **Queues** folder in the **Content** view, if you are not interested the creation date and time of the queues, you can choose not to display those attributes, or you can move those columns to somewhere else in the table.

The changes that you make are saved as schemes so that you can apply the same set of changes to the same type of objects on other queue managers too.

 IBM MQ Explorer supplies and applies standard schemes. Because IBM MQ for z/OS for queue managers and objects can have slightly different attributes, each object scheme has settings for the object on Multiplatform queue managers and for z/OS queue managers. The standard schemes include all the attributes for objects of that type. For example, the **Standard for Queues** scheme includes all the attributes for queues on Multiplatforms and z/OS platforms so that you can be sure that you can see all the attributes for the queues that are listed.

To apply an existing scheme to a table:

1. In the **Content** view, or dialog that contains the table, click the small arrow next to the current scheme name. A menu is displayed.
2. From the menu, click **Select Scheme** The **Select Scheme** dialog opens.
3. In the **Select Scheme** dialog, click the scheme that you want to apply. The attributes that the scheme will display are listed in the dialog.
4. Click **OK**.

The selected scheme is applied to the folder of objects.

Related tasks

[“Creating a scheme” on page 251](#)

You can create schemes for most of the tables of data in IBM MQ Explorer.

[“Editing an existing scheme” on page 252](#)

You can edit any schemes that you have created previously and you can also edit the schemes that are supplied with IBM MQ Explorer, for example, the **Standard for Queues** scheme.

[“Copying an existing scheme” on page 253](#)

If there already exists a scheme that is similar to a scheme that you want to create, you can copy the existing scheme and then edit it as required.

[“Filtering the objects displayed in tables” on page 225](#)

When object data is displayed in IBM MQ Explorer in tables, you can filter the data so that only the objects in which you are interested are displayed.

Creating a scheme

You can create schemes for most of the tables of data in IBM MQ Explorer.

About this task

You can create schemes for queues, channels, and listeners; you can also create schemes for status tables in the **Status** dialogs, such as the Queue Status dialog. A scheme can be applied only to one type of object because different types of objects have different attributes.

Multi The following instructions use an example of creating a scheme for queues so that only the Queue name, Queue type, and Current queue depth attributes are displayed for queues on Multiplatforms.

z/OS The same attributes plus QSG disposition are displayed for queues on z/OS.

You can easily adapt the instructions to create schemes for other types of object too.

To create a scheme, complete the following steps.

Procedure

1. In the **Content** view or dialog that contains the table, click the small arrow next to the current filter name. A menu is displayed.
2. From the menu, click **Manage Schemes**. The **Manage Schemes** dialog opens displaying schemes that already exist for the object type.
3. In the **Manage Schemes dialog**, click **Add**. The **Add Scheme** dialog opens.
4. In the Add Scheme dialog, in the **Scheme Name** field, type a name for the scheme; for example, Monitoring the depth of my queues. By default, all of the attributes are included in the scheme.
5. Edit the scheme as required for distributed objects and for z/OS objects. For example:
 - a) On the **Distributed** page, click **Remove All**. All the attributes in the **Displayed attributes** list are removed.
 - b) In the **Available attributes** list, click **Queue name**, then click **Add**. The Queue name attribute is added to the **Displayed attributes** list.
 - c) Repeat step 6 for the Queue type and Current queue depth attributes.
 - d) Click the z/OS tab to change to the **z/OS** page.
 - e) On the **z/OS** page, click **Copy Distributed to z/OS**. The changes that you made on the **Distributed** page are copied to the **z/OS** page.
 - f) In the **Available attributes** list, click **QSG disposition**, then click **Add**. The QSG disposition attribute is added to the **Displayed attributes** list.
6. Click **OK**. The **Add Scheme** dialog closes. The new scheme is displayed in the **Manage Schemes** dialog along with the other available schemes.
7. Click **OK** to close the **Manage Schemes** dialog.

Results

You can now apply the scheme to a table of data.

Related concepts

[“Define schemes to change the order of columns in tables” on page 250](#)

When object data is displayed in IBM MQ Explorer in tables, you can customized the order of the columns in the tables.

Related tasks

[“Editing an existing scheme” on page 252](#)

You can edit any schemes that you have created previously and you can also edit the schemes that are supplied with IBM MQ Explorer, for example, the `Standard for Queues` scheme.

[“Copying an existing scheme” on page 253](#)

If there already exists a scheme that is similar to a scheme that you want to create, you can copy the existing scheme and then edit it as required.

Editing an existing scheme

You can edit any schemes that you have created previously and you can also edit the schemes that are supplied with IBM MQ Explorer, for example, the `Standard for Queues` scheme.

About this task

You can edit a scheme that is currently applied to the table that you are viewing, or edit another scheme. After modifying the layout of the status table, you can reset the width of the columns to their default values.

Results

The changes to the scheme are automatically applied to any tables that are using that scheme.

Editing the current scheme

About this task

To edit the scheme that is currently applied to the table that you are viewing:

Procedure

1. Make sure that the type of object for which you are creating a scheme is displayed in the **Content** view, then, in the **Content** view, click the small arrow next to the current scheme name. A menu is displayed.
2. From the menu, click **Edit Current Scheme**. The **Edit Scheme** dialog opens.
3. In the **Edit Scheme** dialog, make the changes, then click **OK**. For more information about the dialog, see [Creating a scheme](#).

Editing another scheme

About this task

To edit an existing scheme that is not currently applied to the table that you are viewing:

Procedure

1. Make sure that the type of object for which you are creating a scheme is displayed in the **Content** view, then, in the **Content** view, click the small arrow next to the current scheme name. A menu is displayed.
2. From the menu, click **Manage Schemes**. The **Manage Schemes** dialog opens displaying the schemes that exist for the object type.
3. In the **Manage Schemes** dialog, click the scheme that you want to edit, then click **Edit**. The **Edit Scheme** dialog opens.
4. In the **Edit Scheme** dialog, add or remove attributes from the scheme as required, then click **OK**.

For more information about the dialog, see [Creating a scheme](#).

5. Click **OK** to close the **Manage Schemes** dialog.

Results

The changes to the scheme are automatically applied to any tables that are using that scheme.

Resetting the status table

About this task

To reset the width of the columns to their default values after you have modified the layout of the status table:

Procedure

1. Make sure that the table for which you are resetting the column widths is displayed in the **Content** view, then, in the **Content** view, click the small arrow next to the current scheme name. A menu is displayed.
2. From the menu, click **Reset Column Widths**. The width of the columns on the status table are reset to their default values.

Related concepts

[“Define schemes to change the order of columns in tables” on page 250](#)

When object data is displayed in IBM MQ Explorer in tables, you can customized the order of the columns in the tables.

Related tasks

[“Copying an existing scheme” on page 253](#)

If there already exists a scheme that is similar to a scheme that you want to create, you can copy the existing scheme and then edit it as required.

[“Creating a scheme” on page 251](#)

You can create schemes for most of the tables of data in IBM MQ Explorer.

Copying an existing scheme

If there already exists a scheme that is similar to a scheme that you want to create, you can copy the existing scheme and then edit it as required.

About this task

You can copy any scheme that you have created previously and you can also copy the schemes that are supplied with IBM MQ Explorer; for example, the **Standard for Queues** scheme. You cannot copy a scheme for one type of object to use on another type of object; for example, you cannot copy a scheme for channels to use for filtering queues.

To copy an existing scheme:

Procedure

1. Make sure that the type of object for which you are creating a filter is displayed in the **Content** view, then, in the **Content** view, click the small arrow next to the current filter name. A menu is displayed.
2. From the menu, click **Manage Schemes** The **Manage Schemes** dialog opens displaying the schemes that already exist for the object.
3. In the **Manage Schemes** dialog, click the scheme that you want to copy, then click **Copy As The Copy Scheme** dialog opens.
4. In the **Copy Scheme** dialog, type a name for the new scheme, then click **OK**.
5. In the **Manage Schemes** dialog, click **Edit** The **Edit Scheme** dialog opens.

6. In the **Edit Scheme** dialog, add or remove attributes from the scheme as required, then click **OK**.
7. Click **OK** to close the **Manage Schemes** dialog.

Results

You can now apply the scheme to a table of data.

Related concepts

[“Define schemes to change the order of columns in tables” on page 250](#)

When object data is displayed in IBM MQ Explorer in tables, you can customized the order of the columns in the tables.

Related tasks

[“Editing an existing scheme” on page 252](#)

You can edit any schemes that you have created previously and you can also edit the schemes that are supplied with IBM MQ Explorer, for example, the Standard for Queues scheme.

[“Creating a scheme” on page 251](#)

You can create schemes for most of the tables of data in IBM MQ Explorer.

Changing the colors

You can change the colors that are used to highlight parts of the IBM MQ Explorer interface.

About this task

In IBM MQ Explorer, there are several places where color is used to highlight parts of the interface. For example, in the Content view, cells that are not applicable to an object are colored gray; in the command window that contains the command details of the Create Queue Manager wizard, sections of the text are highlighted in different colors. In the Preferences dialog, you can change the colors that are used.

To change the color of cells that are not applicable:

Procedure

1. Open the Preferences dialog: **Window > Preferences**
2. In navigation tree of the Preferences dialog, expand **MQ Explorer**, then click **Colors**.
3. On the **Colors** page, click the palette button for the feature that you want to change. The palette button in the Content View section of the page controls the color of cells that are not applicable (cells that are colored gray by default); the palette buttons in the Command Details section of the page control the color of the text and background in the command windows that are displayed in the Details window when you create, delete, start, and stop a queue manager in IBM MQ Explorer.
4. In the palette, click the color that you want to use (or define a custom color), then click **OK**.
5. Click **OK** to close the Preferences dialog.

Results

The color that you selected is used.

Related tasks

[“Configuring IBM MQ Explorer” on page 223](#)

Use this information to help you to configure your IBM MQ Explorer installation.

Related reference

[“Accessibility in IBM MQ Explorer” on page 309](#)

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully.

Enabling installed plug-ins

If a new plug-in that you install in IBM MQ Explorer is not enabled by default, you can enable it by using the Preferences dialog.

About this task

When you install a new plug-in in IBM MQ Explorer, whether the plug-in is supplied by IBM or by a third party, if the plugin does not appear to be working in IBM MQ Explorer, it is likely that the plug-in is not enabled by default.

To enable an installed plugin, complete the following steps.

Procedure

1. Click **Window > Preferences** to open the Preferences dialog.
2. In the navigation tree of the **Preferences** dialog, expand **MQ Explorer**, then click **Enable plug-ins**. A list of the available plugins is displayed.
3. Select the check box next to the plug-in that you want to enable, then click **OK**.

Results

The plug-in is now enabled in IBM MQ Explorer. Any folders or menu items for example, that are related to the plug-in are now available in IBM MQ Explorer.

You can also disable plug-ins that you do not use. For example, if you do not use clustering in your messaging networks, you can clear the check box next to the Cluster Component plugin. The Cluster Component plugin remains installed on your computer so that you can enable it in future. Because the plug-in is still installed on your computer, the help that is associated with clustering is still available in the help system and in the context-sensitive help.

Changing the refresh frequency of queue manager information

In IBM MQ Explorer, the information about local and remote queue managers is automatically refreshed at set intervals. You can change the refresh frequency for a specific queue manager, or the default refresh frequency for all new queue managers. You can also prevent a specific queue manager's information from being refreshed automatically.

About this task

By default, the information that is displayed about remote queue managers is refreshed less frequently than the information about local queue managers because of the increase in network traffic every time that the information is requested from remote systems.

You can also prevent specific queue managers being refreshed automatically. For example, if you know that a queue manager will remain stopped for a period of time, you could prevent IBM MQ Explorer requesting information about it and so reduce the network traffic.

- [“Changing the refresh frequency for a specific queue manager” on page 256](#)
- [“Changing the default refresh frequency for all new queue managers” on page 256](#)
- [“Preventing automatic refresh of a queue manager's information” on page 256](#)

Changing the refresh frequency for a specific queue manager

About this task

To change the frequency with which information about a specific queue manager is refreshed:

Procedure

1. In the **Navigator** view, right-click the queue manager, then click **Connection Details > Set Refresh Interval**. The **Automatic Refresh** dialog opens.
2. In the **Automatic Refresh** dialog, edit the value in the **Interval** field.
3. Optional: To reset the automatic refresh rate to the default value, click **Apply Default**.
4. Click **OK** to save the new refresh rate.

Results

The information about the queue manager is now automatically refreshed at the new rate.

Changing the default refresh frequency for all new queue managers

About this task

To change the frequency with which the information about local and remote queue managers is refreshed:

Procedure

1. Click **Window > Preferences** to open the Preferences dialog.
2. On the **MQ Explorer** page, in the **Default Queue Manager Refresh Intervals** fields, type the refresh interval, in seconds, then click **OK**.

Results

All new queue managers that are added to IBM MQ Explorer are now refreshed at the new rate.

Preventing automatic refresh of a queue manager's information

About this task

To prevent queue manager information in IBM MQ Explorer from being refreshed automatically:

Procedure

1. In the **Navigator** view, right-click the queue manager, then click **Connection Details > Set Refresh Interval**. The **Automatic Refresh** dialog opens.
2. In the **Automatic Refresh** dialog, clear the check box, then click **OK**.

Results

The information about the queue manager is no longer refreshed automatically. To refresh the information about the queue manager, click **Refresh** on the menu in the **Content** view.

Specifying the default values used to connect to remote queue managers

You can either configure IBM MQ Explorer with the default values directly, or configure IBM MQ Explorer to use system environment value variables.

About this task

To configure IBM MQ Explorer with default values to use when connecting to remote queue managers, complete one of the following tasks:

- [“Specifying the default values directly” on page 257](#)
- [“Using the system environment variable” on page 257](#)

Specifying the default values directly

About this task

To configure IBM MQ Explorer with the default port number and server-connection channel used to connect to remote queue managers, complete this task in IBM MQ Explorer on the computer from which you want to connect to the remote queue manager.

Procedure

1. In IBM MQ Explorer, click **Window > Preferences**.
The **Preferences** dialog opens.
2. Expand **MQ Explorer**.
3. Expand **Client Connections**.
4. Select **Remote Queue Managers** to display the **Remote Queue Managers** pane.
5. In the **Port number** field, enter the default port number to be used when connecting to a remote queue manager.
6. In the **Server-connection channel** field, enter the default channel name to be used when connecting to a remote queue manager.

Results

IBM MQ Explorer now uses the provided defaults when creating a new connection to a remote queue manager. These defaults can be changed when adding in a new connection to a remote queue manager using the **Add Queue Manager** wizard.

Using the system environment variable

About this task

To configure IBM MQ Explorer to use the system environment variable values when connecting to remote queue managers, complete this task in IBM MQ Explorer on the computer from which you want to connect to the remote queue manager.

Procedure

1. In IBM MQ Explorer, click **Window > Preferences**.
The **Preferences** dialog opens.
2. Expand **MQ Explorer**.
3. Expand **Client Connections**.
4. Select **Remote Queue Managers** to display the **Remote Queue Managers** pane.
5. Select the **Use environment (MQSERVER)** check box to override the specified defaults and use the system environment variable to set the default port and default channel name.

Results

IBM MQ Explorer now uses the MQSERVER system environment variable when creating a new connection to a remote queue manager. These defaults can be changed when adding in a new connection to a remote queue manager using the **Add Queue Manager** wizard.

Related tasks

[“Showing a remote queue manager” on page 99](#)

If you want to administer a remote queue manager, you must connect IBM MQ Explorer to the remote queue manager, so that the queue manager then appears in the Navigator view.

Exporting and importing settings

You can export your settings from IBM MQ Explorer for backup purposes, or to transfer and import the settings into another instance of IBM MQ Explorer.

About this task

You can export and import the following types of settings in IBM MQ Explorer:

- Column schemes that you have created
- Filters that you have created
- Connection details for remote queue managers
- Preferences that you have set in IBM MQ Explorer
- Queue manager set memberships, set definitions, and set filters

Exporting settings

About this task

You can export your settings from your workspace to transfer and import the settings into another instance of IBM MQ Explorer for example.

To export your settings from your IBM MQ Explorer workspace:

Procedure

1. To open the Export dialog, in the Navigator view, right-click **IBM MQ**, then click **Export MQ Explorer settings....** Alternatively, you can open the Export dialog by clicking **File > Export**, then selecting **MQ Explorer > MQ Explorer Settings** from the dialog.
2. Select the check boxes for the types of settings that you want to export.
3. As the data is written to the file system, enter the file name and location for the compressed file that will be created to store the exported settings.
4. Click **OK**.

Results

A compressed file that contains the exported settings is created. The compressed file contains the settings in XML files.

For information about exporting queue manager sets, see: [“Importing and exporting queue manager sets” on page 249](#).

Importing settings

About this task

To import settings to IBM MQ Explorer:

Procedure

1. In the Navigator view, right-click **IBM MQ**, then click **Import MQ Explorer settings...** The Import dialog opens.
2. Browse for the compressed file that contains the settings.

3. Select the types of settings that you want to import into IBM MQ Explorer. If the compressed file does not contain settings of a certain type, the check box associated with that type is unavailable.
4. Click **OK**.

Results

The settings from the compressed file are imported into IBM MQ Explorer.

For information about importing queue manager sets, see: [“Importing and exporting queue manager sets” on page 249](#).

Including SYSTEM objects when you run tests

By default, SYSTEM objects are not included in test results, but you can choose to include them if required.

About this task

Definitions of SYSTEM.DEFAULT objects are provided in IBM MQ as incomplete templates so, by default, they are not included when you run tests. However, you can include them if you want.

To include SYSTEM objects in test results, complete the following steps.

Procedure

1. Click **Window > Preferences** to open the Preferences dialog.
2. In the navigation tree of the Preferences dialog, expand **IBM MQ Explorer**, then click **Tests**.
3. Select the **Include SYSTEM objects in the test results** check box.

Results

Next time you run tests against objects in IBM MQ Explorer, any available SYSTEM objects are also tested.

Including hidden queue managers in test configurations

Queue managers that are hidden in IBM MQ Explorer do not by default appear in the list of available objects when you create new test configurations. However, you can choose to have hidden queue managers included so that they listed as available queue managers against which you can run tests.

About this task

Queue managers that are currently hidden in IBM MQ Explorer are not currently of interest to you so, by default, they are not included in the list of available objects when you create new test configurations.

To include hidden queue managers, complete the following steps.

Procedure

1. Click **Window > Preferences** to open the Preferences dialog.
2. In the navigation tree of the Preferences dialog, expand **IBM MQ Explorer**, then click **Tests**.
3. Select the **Include hidden objects in the list of available objects** check box.

Results

Next time you create or edit a test configuration, any hidden queue managers are listed as available queue managers against which you can run the tests.

Displaying object authority settings as text

In IBM MQ Explorer dialogs, object authorities are displayed as icons by default, but you can choose to have them displayed as text instead.

About this task

In the Manage Authority Records dialogs, and in other dialogs that display object authorities, the tables use icons to show whether an authority is granted. If you prefer, you can set the tables to use text instead of icons.

To change the tables to use text instead of icons to show whether an authority is granted, complete the following steps.

Procedure

1. Open the Preferences dialog: **Window > Preferences**
2. Expand **MQ Explorer**.
3. On the **Authorization Service** page, click **Display authorities as text**.
4. Click **OK** to close the Preferences dialog.

Results

The next time that you open a dialog that displays object authorities, the tables will show authorities using text instead of icons.

Related tasks

[“Configuring IBM MQ Explorer” on page 223](#)

Use this information to help you to configure your IBM MQ Explorer installation.

Related reference

[“Accessibility in IBM MQ Explorer” on page 309](#)

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully.

Advanced Message Security

Advanced Message Security is a separately installed and licensed component that provides a high level of protection for sensitive data flowing through the IBM MQ network, while not impacting the end applications

Message signing

By using a digital signature on the message the identity of the sender and the authenticity of the message can be confirmed, and therefore the sender of the message is unable to deny (or repudiate) the sending of that message.

When an application places a message on a queue, Advanced Message Security checks if the target queue has a Advanced Message Security policy for signing or encryption. If signing is required, Advanced Message Security creates an envelope containing the message data, a cryptographic signature, and the public certificate data of the user associated with the application.

When an application retrieves the message from the queue, Advanced Message Security strips the signature from the message data and verifies that the sender is known and signed by a trusted certificate authority. In addition, Advanced Message Security checks that the user identified by the signature is authorized, by policy, to place messages on the target queue.

The signature also includes a digest of the message data, generated at the time the message was placed on the queue. This digest is verified to ensure that the data in the message has not been altered between being placed on the queue and being retrieved.

Message encryption

By using message encryption, a message sender can be sure that the content of the message has not been modified before reaching the recipient.

When an application places a message on a queue, Advanced Message Security checks if the target queue has a Advanced Message Security policy for signing or encryption. If encryption is required, Advanced Message Security signs and encrypts the data.

In addition to the signing process, Advanced Message Security encrypts the message data with a symmetric key, using the encryption algorithm specified in the Advanced Message Security policy associated with the target queue. The message is then addressed to each potential recipient specified in that policy, using the users' public keys.

When an application retrieves the message from the queue, Advanced Message Security verifies the signature and decrypts the message data using the private key of the recipient user.

Distinguished names

Advanced Message Security uses the Public Key Infrastructure (PKI) identity to represent a user or an application. This type of identity is used for signing and encrypting messages. The identity is represented by the distinguished name (DN) field in a certificate associated with signed and encrypted messages.

Sender distinguished names

The sender distinguished names (DNs) identify users authorized to place messages on a queue. However, Advanced Message Security does not check whether a message has been placed on a data-protected queue by a valid user until the message is retrieved. At this time, if the policy stipulates one or more valid senders, and the user that placed the message on the queue is not in the list of valid senders, Advanced Message Security returns an error to the getting application, and place the message on its error queue.

A policy can have 0 or more sender DNs specified. If no sender DNs are specified for the policy, any user can put data-protected messages to the queue providing the user's certificate is trusted.

Sender distinguished names have the following form:

```
CN=Common Name,O=Organization,C=Country
```

If one or more sender DNs are specified for the policy, only those users can put messages to the queue associated with the policy.

Sender DNs, when specified, must match exactly the DN contained in the digital certificate associated with user putting the message.

Recipient distinguished names

The recipient distinguished names (DN) identify users authorized to retrieve messages from a queue. A policy can have zero or more recipient DNs specified. Recipient distinguished names have this form:

```
CN=Common Name,O=Organization,C=Country
```

If no recipient DNs are specified for the policy, any user can get messages from the queue associated with the policy. This implies that the policy does not specify encryption, as a policy with encryption requires recipient DNs to be specified.

If one or more recipient DNs are specified for the policy, only those users can get messages from the queue associated with the policy.

Recipient DNs, when specified, must match exactly the DN contained in the digital certificate associated with user getting the message.

Configuring Advanced Message Security policies involves creating the policies using tools provided with Advanced Message Security.

Note: Advanced Message Security does not allow policies for SYSTEM queues. These are queues with a name that begin with 'SYSTEM.'. If you define a policy for a SYSTEM queue, it is ignored.

Troubleshooting

Troubleshooting is the process of finding and eliminating the cause of a problem. If you are having problems with IBM MQ Explorer, use the techniques described to help you diagnose and solve them.

You can prevent many problems with your object definitions and messaging configurations by testing your object definitions for potential problems. For more information, see [Testing your object definitions for problems](#).

If an error message is displayed while you are working in IBM MQ Explorer, click **More Details** on the error message to display more information about the problem.

Related tasks

[“Using IBM MQ Explorer trace” on page 262](#)

You can use IBM MQ Explorer trace to find out what happened to cause a problem.

[“Using IBM MQ Explorer trace in other Eclipse environments” on page 263](#)

IBM MQ Explorer can be installed into your own Eclipse environment or Eclipse-based product to help develop Java applications, or to develop your own extensions.

[“Using IBM MQ trace” on page 268](#)

IBM MQ trace enables you to collect detailed information about what IBM MQ is doing.

Using IBM MQ Explorer trace

You can use IBM MQ Explorer trace to find out what happened to cause a problem.

About this task

Note that you normally enable tracing only when asked to do so by your IBM Service Representative. Tracing slows down IBM MQ Explorer and the trace files can rapidly become large.

When you start the IBM MQ Explorer by using the **runwithtrace** command, extra parameters are incorporated that enable tracing of IBM MQ Explorer. Note the **runwithtrace** command takes no parameters itself.

The output location of the trace file is printed to the command line when the command runs because the actual location:

- Is dependent upon the status of the IBM MQ installation
- User permissions

The IBM MQ Explorer trace files have a name of the format AMQYYYYMMDDHHmmssmmm.TRC.n that indicates the start time of the trace session. When a trace file reaches its maximum size, the trace facility renames all trace files from the session by incrementing the file suffix .n by one. The trace facility then creates a new file with the suffix .0 that contains the latest trace.

To trace IBM MQ Explorer, complete the following steps.

Procedure

1. Close IBM MQ Explorer.

2.  **Windows**

On Windows systems, use **runwithtrace.cmd** to run IBM MQ Explorer with tracing activated.

The **runwithtrace** command is in one of the following locations:

- If you are running the IBM MQ Explorer that was installed as part of a full IBM MQ server installation, the **runwithtrace** command is in `MQ_INSTALLATION_PATH\MQExplorer\Eclipse`, where `MQ_INSTALLATION_PATH` is the IBM MQ installation path.

- If you installed the IBM MQ Explorer from the MS0T SupportPac, the **runwithtrace** command is in the same directory as the **MQExplorer** command.

3. Linux

On Linux systems, use **runwithtrace** to run IBM MQ Explorer with tracing activated.

The **runwithtrace** command is in one of the following locations:

- If you are running the IBM MQ Explorer that was installed as part of a full IBM MQ server installation, the **runwithtrace** command is in `/opt/mqm/mqexplorer/eclipse`, where `opt/mqm` is the IBM MQ installation directory.
- If you installed the IBM MQ Explorer from the MS0T SupportPac, the **runwithtrace** command is in the same directory as the **MQExplorer** command.

Related tasks

[“Using IBM MQ Explorer trace in other Eclipse environments” on page 263](#)

IBM MQ Explorer can be installed into your own Eclipse environment or Eclipse-based product to help develop Java applications, or to develop your own extensions.

[“Installing into Eclipse environments” on page 24](#)

Install IBM MQ Explorer into your own Eclipse environment or Eclipse-based product to help with developing Java applications or to develop your own extensions.

Using IBM MQ Explorer trace in other Eclipse environments

IBM MQ Explorer can be installed into your own Eclipse environment or Eclipse-based product to help develop Java applications, or to develop your own extensions.


About this task

You can collect IBM MQ Explorer trace if you have installed IBM MQ Explorer into your own Eclipse or Eclipse-based product by following the procedure.

Note that you normally enable tracing only when asked to do so by your IBM Service Representative. Tracing slows down IBM MQ Explorer and the trace files can rapidly become large.

Procedure

1. The IBM MQ Explorer trace mechanism relies on AspectJ and Equinox Weaving plug-ins being installed. To confirm that they are installed:
 - a) Click **Help**
 - b) Click **About...**
 - c) Click **Installation Details**
 - d) Click the **Plug-ins** tab.

 The `org.eclipse.equinox.weaving.caching.j9` plug in no longer exists, but you require this plug in if you are using IBM MQ 9.0 Long Term Support, or IBM MQ 9.0 Continuous Delivery releases, prior to IBM MQ 9.0.4.

Verify that the following plug-ins are installed:

```
org.aspectj.runtime
org.aspectj.weaver

org.eclipse.equinox.weaving.aspectj
org.eclipse.equinox.weaving.caching
org.eclipse.equinox.weaving.caching.j9
org.eclipse.equinox.weaving.hook
```

2. If they are not already installed, install the AspectJ and Equinox Weaving plug-ins. These plug-ins must match the version of Eclipse you are using and can be downloaded from the Eclipse AspectJ

Development Tools download site. To determine which download site to use for your version of Eclipse, see <https://projects.eclipse.org/projects/tools.ajdt>.

V 9.0.4 If you are using Eclipse 4.4.2 then you need to use AJDT 2.2.4; do not use AJDT 2.2.3. This statement applies if you are using IBM MQ 9.0 Long Term Support, or IBM MQ 9.0 Continuous Delivery releases, prior to IBM MQ 9.0.4.

IBM MQ 9.0.4 uses Eclipse 4.6.3. If you are using Eclipse 4.6.3, you must obtain a build from the AJDT site which is compatible with Eclipse 4.6.

Currently, these builds are available as development builds only; you should select the latest one available.

To install the AspectJ and Equinox Weaving plug-ins, complete the following substeps:

- a) Click **Help** then click **Install New Software...**
 - b) Click **Add...** then enter the location of the download site appropriate for the Eclipse version. For example, for Eclipse 4.4, enter <http://download.eclipse.org/tools/ajdt/44/dev/update>.
 - c) Click **OK**
A list of available software is displayed.
 - d) Expand the **Other AJDT Tools (Optional)** category and select **AspectJ Compiler and Equinox Weaving SDK**.
 - e) Click **Next** and follow the wizard instructions.
3. To trace IBM MQ Explorer, close and restart the IBM MQ Explorer by using the **runwithtrace** script. For the script to use for Windows operating systems, see [“runwithtrace command script for Windows” on page 265](#). For the script to use for Linux operating systems, see [“runwithtrace command script for Linux” on page 266](#).

The script must be run from the directory that contains the Eclipse executable file.

When you start the IBM MQ Explorer by using the **runwithtrace** command, extra parameters are incorporated that enable tracing of IBM MQ Explorer. Note the **runwithtrace** command takes no parameters itself.

The output location of the trace file is printed to the command line when the command runs because the actual location:

- Is dependent upon the status of the IBM WebSphere MQ installation
- User permissions

The IBM MQ Explorer trace files have a name of the format AMQYYYYMMDDHHmmssmmm.TRC.n that indicates the start time of the trace session. When a trace file reaches its maximum size, the trace facility renames all trace files from the session by incrementing the file suffix .n by one. The trace facility then creates a new file with the suffix .0 that contains the latest trace.

4. Uninstall the AspectJ and Equinox Weaving plug-ins. When you finish with IBM MQ Explorer trace, you can remove the AspectJ and Equinox Weaving plugins from the Eclipse environment. To uninstall AspectJ and Equinox Weaving plug-ins:
 - a) Click **Help** and then click **About...**
 - b) Click **Installation Details** and then click the **Installed Software** tab.
 - c) Select the items AspectJ Compiler and Equinox Weaving SDK.
 - d) Click **Uninstall...** and follow the wizard instructions.

Related tasks

[“Using IBM MQ Explorer trace” on page 262](#)

You can use IBM MQ Explorer trace to find out what happened to cause a problem.

[“Installing into Eclipse environments” on page 24](#)

Install IBM MQ Explorer into your own Eclipse environment or Eclipse-based product to help with developing Java applications or to develop your own extensions.

runwithtrace command script for Windows

To trace MQ Explorer, close and restart the MQ Explorer by using the **runwithtrace** command script. A copy of this script for use on Windows systems is embedded in this topic.

Copy and save the following script as a text file called `runwithtrace.cmd`, then run the script as described in [step 3 of "Using IBM MQ Explorer trace in other Eclipse environments"](#).

```
@echo off

REM -----
REM File Name : runwithtrace.cmd
REM
REM File Description : This script is used when MQ Explorer plug-ins are
REM installed into another Eclipse or Eclipse based product.
REM It launches eclipse and will run WebSphere MQ Explorer with trace enabled.
REM -----

setlocal

REM -----
REM Special case for when MQ Explorer plug-ins are installed in an Eclipse or an
REM Eclipse based product.
REM
REM eclipse needs to be in current directory.
REM -----

if exist "eclipse.exe" goto :MQExplorer_found
goto :no_MQExplorer

:MQExplorer_found
set explorerCmd=eclipse.exe

REM -----
REM Special processing for enabling trace
REM 1. Allow a user to supply their own properties file, pointed to by the
REM    MQPROPERTIES environment variable
REM 2. Otherwise, build a properties file in %temp% which writes trace
REM    to the MQ_INSTALLATION_PATH\trace directory if writeable, otherwise to
REM    %temp% itself
REM -----

if not "%MQPROPERTIES%."=="." goto :own_properties

REM Create a properties file with the default trace options
set MQPROPERTIES=%temp%\mq_trace.properties

REM Set MQTRACE to temp or the MQ trace directory
if "%MQ_JAVA_DATA_PATH%."=="." goto :set_to_temp

set MQTRACE=%MQ_JAVA_DATA_PATH%\trace
goto :finish_set

:set_to_temp
set MQTRACE=%temp%

:finish_set

REM -----
REM Where should trace be written to - Try the MQ trace directory first
REM -----
if "%MQTRACE%"=="%MQ_JAVA_DATA_PATH%\trace" goto :MQ_dir_available
echo Trace will be written to the temporary directory %MQTRACE%
goto :finish_trace_location

:MQ_dir_available
echo Confirming write access to the MQ trace directory %MQTRACE%
echo Test >> "%MQTRACE%\test.gui" 2>NUL
if exist "%MQTRACE%\test.gui" goto :MQ_dir_used
echo Trace will be written to the temporary directory %temp%
set MQTRACE=%temp%
goto :finish_trace_location
```

```

:MQ_dir_used
echo Trace will be written to the MQ trace directory %MQTRACE%
del "%MQTRACE%\test.gui" >nul 2>&1

:finish_trace_location

REM Convert back slashes to forward slashes for use in properties file
REM Note :\: / converts back slashes to forward slashes.
set MQTRACE=%MQTRACE:\=/%

REM -----
REM Now build the default properties file
REM -----
echo Diagnostics.MQ=enabled > %MQPROPERTIES%
echo Diagnostics.Java=all >> %MQPROPERTIES%
echo Diagnostics.Java.Trace.Detail=high >> %MQPROPERTIES%
echo Diagnostics.Java.Trace.Destination.File=enabled >> %MQPROPERTIES%
echo Diagnostics.Java.Trace.Destination.Console=disabled >> %MQPROPERTIES%
echo Diagnostics.Java.Trace.Destination.Pathname=%MQTRACE% >> %MQPROPERTIES%
echo Diagnostics.Java.FFDC.Destination.Pathname=%MQTRACE% >> %MQPROPERTIES%
echo Diagnostics.Java.Errors.Destination.Filename=%MQTRACE% >> %MQPROPERTIES%

:own_properties

REM -----
REM Build the command line
REM All parameters passed to this script are passed through.
REM Set the load time weaving options, it's set as part of the vmargs parameter.
REM -----

REM Note.
REM In eclipse and eclipse based products the osgi.framework.extensions is set
REM as part of the Equinox Weaving plug-ins eclipse installation.
REM Therefore unlike in the normal MQ Explorer script LTW_OPTIONS is empty

REM set LTW_OPTIONS=-Dosgi.framework.extensions=org.eclipse.equinox.weaving.hook
set LTW_OPTIONS=
set explorerCmd=%explorerCmd% %* -vmargs -Xmx512M %LTW_OPTIONS% "-
Dcom.ibm.mq.commonservices=%MQPROPERTIES%"

REM -----
REM Launch MQ Explorer
REM -----
echo Launching %explorerCmd%
start %explorerCmd%

goto :end

:no_MQExplorer
echo ERROR - eclipse.exe not found in the current directory.
echo ERROR - This script needs to be run in the same directory as eclipse.exe

:end
endlocal

```

runwithtrace command script for Linux

To trace MQ Explorer, close and restart the MQ Explorer by using the **runwithtrace** command script. A copy of this script for use on Linux systems is embedded in this topic.

Copy and save the following script as a text file called `runwithtrace.cmd`, then run the script as described in [step 3 of "Using IBM MQ Explorer trace in other Eclipse environments"](#).

```

#!/bin/sh
#-----
# File Name : runwithtrace
#
# File Description : This script is used when MQ Explorer plug-ins are
# installed into another Eclipse or Eclipse based product.
# It launches eclipse and will run WebSphere MQ Explorer with trace enabled.
#
#-----
#

```

```

# Special processing for enabling trace
# 1. Allow a user to supply their own properties file, pointed to by the
#    MQPROPERTIES environment variable
# 2. Otherwise, build a properties file in /tmp which writes trace
#    to /var/mqm/trace directory if writeable, otherwise to /tmp itself
# -----

# test if variable is not set or refers to a file that does not exist
if [ -z "$MQPROPERTIES" -o ! -f "$MQPROPERTIES" ]
then
    # Create a properties file with the default trace options
    MQPROPERTIES=/tmp/mq_trace.properties
    # -----
    # Where should trace go - Try the trace directory first
    # -----
    echo "Confirming write access to the MQ trace directory /var/mqm/trace"
    MQTRACE=/var/mqm/trace
    # test if dir exists and is writable
    if [ -d $MQTRACE -a -w $MQTRACE ]
    then
        echo "Trace will be written to the MQ trace directory /var/mqm/trace"
    else
        echo "Trace will be written to the temporary directory /tmp"
        MQTRACE=/tmp
    fi

    # -----
    # Now build the default properties file
    # -----
    echo Diagnostics.MQ=enabled > $MQPROPERTIES
    echo Diagnostics.Java=all >> $MQPROPERTIES
    echo Diagnostics.Java.Trace.Detail=high >> $MQPROPERTIES
    echo Diagnostics.Java.Trace.Destination.File=enabled >> $MQPROPERTIES
    echo Diagnostics.Java.Trace.Destination.Console=disabled >> $MQPROPERTIES
    echo Diagnostics.Java.Trace.Destination.Pathname=$MQTRACE >> $MQPROPERTIES
    echo Diagnostics.Java.FFDC.Destination.Pathname=$MQTRACE >> $MQPROPERTIES
    echo Diagnostics.Java.Errors.Destination.Filename=$MQTRACE >> $MQPROPERTIES

fi

# -----
# Build the command line to run
# Look in the current directory
# All parameters passed to this script are passed through.
# Set the load time weaving options, it's set as part of the vmargs parameter.
# -----

# Special case for when MQ Explorer plug-ins are installed in an Eclipse or an
# Eclipse based product.
# eclipse needs to be in current directory.

if [ -f "eclipse" ]
then
    explorerCmd="./eclipse"
fi

if [ ! -f "${explorerCmd}" ]
then
    echo "ERROR - eclipse executable could not be found in the current directory"
    echo "ERROR - This script needs to be run in the same directory as the eclipse executable"
    exit 1
fi

# Note.
# In eclipse and eclipse based products the osgi.framework.extensions is set
# as part of the Equinox Weaving plug-ins eclipse installation.
# Therefore unlike in the normal MQ Explorer script LTW_OPTIONS is empty

# LTW_OPTIONS=-Dosgi.framework.extensions=org.eclipse.equinox.weaving.hook
# LTW_OPTIONS=
explorerCmd="$explorerCmd $* -vmargs -Xmx512M $LTW_OPTIONS
-Dcom.ibm.mq.commonservices=$MQPROPERTIES"

# -----
# Launch MQ Explorer
# -----
echo Launching $explorerCmd

exec $explorerCmd

```

Using IBM MQ trace

IBM MQ trace enables you to collect detailed information about what IBM MQ is doing.

About this task

You normally enable tracing only when asked to do so by your IBM service representative. Tracing slows down IBM MQ and the trace files can rapidly become very large.

For more information, see [strmqtrc](#) and [endmqtrc](#) in the IBM MQ online product documentation.

Using IBM MQ Explorer to start and stop tracing is equivalent to using the control commands **strmqtrc -e** and **endmqtrc -e**, which trace all processes on the specified queue manager.

IBM MQ trace does not trace IBM MQ Explorer. For details about how to trace IBM MQ Explorer, see [“Troubleshooting” on page 262](#).

Starting trace

About this task

To turn on the trace service:

Procedure

1. In the Navigator view, right-click **IBM MQ**, then click **Trace....**
2. In the Trace dialog, select one or more of the following options:
 - To output data for every trace point in the system, click **All**.
 - To activate tracing at high-detail level for flow processing trace points, click **Detail**.
3. Click **Start**.

Results

The IBM MQ trace starts writing information to the trace files. IBM MQ continues to write to the trace files until you stop the trace.

Stopping trace

About this task

To turn off the trace service:

Procedure

1. In the Navigator view, right-click **IBM MQ**, then click **Trace....**
2. Click **Stop**.

Results

The IBM MQ trace stops writing to the trace files.

Viewing trace files

About this task

You can find the local trace files in the `trace` subdirectory of the directory that was chosen during the installation for data files. If you installed to the default directories, the `trace` directory is in the IBM MQ program installation directory.

The files in the trace directory have names like AMQ123 .TRC where 123 is the PID of the process that wrote the file. You can view the files using any viewer that can display a simple ASCII file.

MQ Telemetry

MQ Telemetry supports the connection of an IBM MQ message server to telemetry devices including, but not limited to, sensors and actuators, mobile phones, smart meters, medical devices, vehicles, and satellite locations. The connection is made possible by the MQ Telemetry Transport (MQTT) protocol.

MQTT is an open message format and protocol that you can use to transfer messages from telemetry devices to a message server or the other way round. It can run on constrained devices and across constrained networks. Constraints on devices include low memory and processing capability. Constraints on networks include low bandwidth, high latency, high cost, and fragility. MQTT has been successfully deployed in various industries, including energy, utilities, and the retail sector.

Related information

[Developing applications for MQ Telemetry](#)

[MQ Telemetry security](#)

MQ Telemetry objects

This information provides details on MQ Telemetry objects which include: telemetry channels, telemetry channel status objects, and the MQXR service.

Related concepts

[“Telemetry \(MQXR\) service” on page 269](#)

The IBM MQ Extended Reach (MQXR) service is more commonly referred to as the MQ Telemetry service. It is a TCP/IP listener that is installed as an IBM MQ service. It runs when a queue manager starts or stops.

[“Telemetry channels” on page 270](#)

A Telemetry channel is a communication link between a queue manager on IBM MQ, and MQTT clients. Each channel might have one or more telemetry devices connected to it.

[“Telemetry channel status objects” on page 270](#)

A telemetry channel status object is an MQTT client that collects information from telemetry devices attached to it and sends the information to IBM MQ.

Telemetry (MQXR) service

The IBM MQ Extended Reach (MQXR) service is more commonly referred to as the MQ Telemetry service. It is a TCP/IP listener that is installed as an IBM MQ service. It runs when a queue manager starts or stops.

An MQXR service is defined when you run the **Define sample configuration** wizard. Only one instance of this service can be defined per queue manager.

To view the MQXR service, click the **Services** folder on the queue manager to which the service belongs, in the Navigator view. Ensure that the **Show System Objects** option is selected, and navigate to the service. The service is called `SYSTEM.MQXR.SERVICE`.

You can modify the properties of an MQXR service like a standard IBM MQ service.

Related tasks

[“Defining the MQXR service” on page 278](#)

The MQXR service is defined when you run the **Define sample configuration** wizard. You can also define the MQXR service manually.

[“Starting and stopping the MQXR service” on page 280](#)

Before you can start or stop the MQXR service, the queue manager must be running.

Telemetry channels

A Telemetry channel is a communication link between a queue manager on IBM MQ, and MQTT clients. Each channel might have one or more telemetry devices connected to it.

For messages flowing from IBM MQ to MQTT clients, messages are taken from the default MQTT transmit queue, and sent through the telemetry channel. Messages destined for specific MQTT clients are routed to them using their client identifiers.

Advanced option

Telemetry channels have an option which sets the maximum number of client connections that can be displayed in the **Channel Status Content** view. This option is called **Max responses**. The default value is 500. Consider configuring this option before you start your queue manager. If your queue manager is running, you must restart it to apply the advanced option changes.

To configure the maximum responses option, perform the following actions:

1. Click **Window > Preferences**.
2. Expand **IBM MQ Explorer**, then click **Telemetry**.
3. In the **Max responses** field, type the number of client connections to display at any one time.
4. Click **OK**.

Client connections on all telemetry channels up to the maximum response limit are shown in the **Channel Status Content** view. If client connections exceed this limit, a warning is displayed within the **Content** view. For example, if you set the maximum responses to 10 and you reach or exceed this number, the following warning is displayed: The display has been limited to the first 10 responses. Use a filter to select a subset of responses.

The **Telemetry channel status window** shows client connections specific to that channel. The maximum response option limit applies only to client connections on this channel.

Related tasks

[“Creating and configuring a telemetry channel” on page 275](#)

A telemetry channel connects a number of MQTT clients to IBM MQ. Create one or more telemetry channels on a queue manager. Each of these telemetry channels might have different configuration settings, making it easier to manage the clients attached to them.

[“Starting and stopping a telemetry channel” on page 280](#)

[“Viewing the status of a telemetry channel” on page 281](#)

[“Filtering Telemetry objects” on page 282](#)

If you are viewing several defined telemetry objects in the **Content** view, you might need a way to narrow the search scope of these objects. Do this by using filters.

Telemetry channel status objects

A telemetry channel status object is an MQTT client that collects information from telemetry devices attached to it and sends the information to IBM MQ.

MQTT clients appear as queue managers to other standard IBM MQ queue managers. In the same way that channels are a communication link between queue managers in IBM MQ, a telemetry channel performs the same purpose. It links a queue manager to MQTT clients.

You can write your own MQTT client application that implements the MQTT v3 protocol. For more information about writing MQTT client applications, see [Developing applications for IBM MQ Telemetry](#) in the IBM MQ online product documentation.

Purge an MQTT client

Purging an MQTT client connection disconnects the client from a telemetry channel and cleans up the state of that client.

Cleaning the state of a client involves deleting all pending publications, and removing all subscriptions from that client.

Related tasks

[“Viewing the status of a telemetry channel” on page 281](#)

[“Filtering Telemetry objects” on page 282](#)

If you are viewing several defined telemetry objects in the **Content** view, you might need a way to narrow the search scope of these objects. Do this by using filters.

Related reference

[“Telemetry channel status attributes” on page 288](#)

As with IBM MQ, you can view the status of a telemetry channel. For each attribute, there is a brief description of what information the attribute is used for. All of the telemetry channel status attributes are read-only.

MQTT client utility

The MQTT client utility is a Java application with which you can explore the features of MQTT, that is, connecting to a queue manager, and publishing and subscribing to topics.

You can use the client utility to verify your telemetry setup (for example, when you create a telemetry channel), or the client might serve as an aid when you are developing and debugging applications. For more information about the features and properties of the MQTT client utility, see the following descriptions:

Publish

Publish a message to a topic which is then distributed to interested subscribers.

Subscribe

Clients and the MQXR listener can register their interest in a topic by subscribing to it and thus receive messages published to that topic.

Topic

A topic is the key that identifies the information channel to which messages are published. Subscribers use the topic name to identify the information channels on which they want to receive published messages.

Topic wildcards

A subscription might contain special characters, and you can subscribe to multiple topics at once. A topic level separator uses the forward slash (/) to separate each level within a topic, creating a hierarchical structure. For example, `ibm/qmgr/apple`. Topic level separators allow more flexibility and simplifies management of topics.

For subscriptions, two wildcard characters are supported:

- The number sign (#) is used to match any number of levels within a topic. For example, if you subscribe to the topic `ibm/qmgr/#`, you receive messages on the topics `ibm/qmgr/apple` and `ibm/qmgr/orange`.
- The plus sign (+) matches only one topic level. For example, `ibm/qmgr/+` matches `ibm/qmgr/apple` but not `ibm/qmgr/apple/queue`.

You can use the + at the end of the topic tree or within the topic tree. For example, `ibm/+` and `ibm/+/orange` are both valid.

Publication topics are not allowed to contain either the + or #.

Message

A message, in the context of an MQTT client utility, refers to the message payload distributed to subscribers that show interest in a topic. Messages can consist of alphanumeric characters.

Quality of service (QoS)

The MQTT client utility provides three qualities of service, as follows:

At most once (QoS = 0)

The message is delivered at most once, or it is not delivered at all. There is no acknowledgment on receipt of the message.

The message might be lost if the client is disconnected, or if the server fails.

At least once (QoS = 1)

The message is delivered at least once. It might be delivered multiple times if no acknowledgment is received after a period of time, or if a failure is detected and the communication session is restarted.

The message must be stored locally at the sender until an acknowledgment is received, in case the message must be sent again. Messages might be duplicated at the receiving application.

Exactly once (QoS = 2)

This is the highest level of delivery where a message is delivered once and only once. Delivery is assumed, but duplicate messages are not delivered to the receiving application.

Retained

This option is only used when publishing messages. It determines whether a message, from a publication on a topic, is retained by the MQTT server (in IBM MQ, the queue manager) after it is delivered to the current subscribers. If you create a subscription to a topic that has a retained publication, you immediately receive the most recent retained publication on that topic.

Last will and testament

This is a connection option which determines the message that is sent to IBM MQ, in the event of the MQTT client being disconnected unexpectedly. This option includes a topic, message, QoS, and the option to retain the publication. Consider setting the **QoS** to 1 or 2 to guarantee delivery.

Clean session

Starting an MQTT client with a clean session removes all pending publications, and all old subscriptions that existed on the client before connection. If no previous session exists, the client utility starts with a new session.

Client history

The client history provides information about events that occur while using the MQTT client utility. Examples of events include: Connected, Disconnected, Published, or Subscribed.

View the full details of an entry by selecting the entry and pressing **Enter**. Alternatively, double-click the entry.

Reorder the columns in the client history by dragging the column names and placing them in any order of your choice.

Related tasks

[“Running the MQTT client utility” on page 273](#)

You can run the MQTT client utility in several ways. You can run it when you create a new telemetry channel, when you run the **Define sample configuration** wizard, from an already existing telemetry channel, or from the MQ Telemetry welcome page.

[“Using the MQTT client utility” on page 274](#)

Use one or more MQTT client utilities to test your telemetry setup. Within the scope of this task, one client utility is used to publish and subscribe to messages.

Running the MQTT client utility

You can run the MQTT client utility in several ways. You can run it when you create a new telemetry channel, when you run the **Define sample configuration** wizard, from an already existing telemetry channel, or from the MQ Telemetry welcome page.

About this task

When you launch the MQTT client utility after a basic configuration is set up or a new telemetry channel is created, the utility retains the port number and host name of the telemetry channel.

Methods to launch the client utility

Running the utility from the Define sample configuration wizard

Follow the steps for defining a sample configuration. For more information, see [“Defining a sample configuration” on page 277](#).

1. When the **Define sample configuration** wizard starts, select **Launch MQTT Client Utility**.
2. Click **Finish**.

Running the utility from the New Telemetry Channel wizard

Follow the steps for creating a new telemetry channel using the wizard. For more information, see [“Creating and configuring a telemetry channel” on page 275](#).

1. On the **Summary page** of the wizard, select **Launch MQTT Client Utility**.
2. Click **Finish**.

Running the utility from the IBM MQ Telemetry Welcome page

1. Click the **Telemetry** folder to display the MQ Telemetry welcome page.
2. From the **Content** view, click **Run MQTT Client Utility**.

Running the utility from a telemetry channel

You can launch the MQTT client utility on specific telemetry channels.

1. Expand the **Telemetry** folder, then click **Channels**. Your telemetry channels are listed in the **Content** view.
2. Right-click a telemetry channel and select **Launch MQTT Client Utility**.

Note: If you choose to perform authentication with either JAAS or TLS, you do not get the option to launch the MQTT client utility from a telemetry channel. This is because the MQTT client utility does not support JAAS or TLS authentication. However, you can write your own MQTT client application to support authentication using JAAS or TLS.

Related tasks

[“Defining a sample configuration” on page 277](#)

You can use the **Define sample configuration** wizard to reconfigure your queue manager, making it suitable for the MQ Telemetry feature. The sample configuration defines and starts the MQXR service, defines the transmit queue, and creates a sample telemetry channel.

[“Creating and configuring a telemetry channel” on page 275](#)

A telemetry channel connects a number of MQTT clients to IBM MQ. Create one or more telemetry channels on a queue manager. Each of these telemetry channels might have different configuration settings, making it easier to manage the clients attached to them.

[“Using the MQTT client utility” on page 274](#)

Use one or more MQTT client utilities to test your telemetry setup. Within the scope of this task, one client utility is used to publish and subscribe to messages.

Using the MQTT client utility

Use one or more MQTT client utilities to test your telemetry setup. Within the scope of this task, one client utility is used to publish and subscribe to messages.

Before you begin

- Make sure that the telemetry (MQXR) service is running.
- You need at least one running telemetry channel to successfully use the MQTT client utility.

About this task

There are several methods to start the MQTT client utility. For more information about running the client utility, see [“Running the MQTT client utility” on page 273](#). Within the scope of this task, the client utility is started from the PlainText telemetry channel created with the **Define sample configuration** wizard.

By default, the **MCA user ID** of the PlainText channel is set to guest on Windows, and nobody on Linux. You need these default values in order to subscribe to topics.

Procedure

1. Right-click the PlainText telemetry channel, then click **Run MQTT Client Utility**. The client utility window opens. The **Host** and **Port** fields are automatically set using values from the selected telemetry channel.
2. Type a client ID in the **Client identifier** field. A new client identifier is generated each time you launch an MQTT client utility from a telemetry channel. You can either use the generated identifier or type a name of your choice. If you run more than one client utility on a telemetry channel, ensure that you use different client IDs for each client utility. If two MQTT client utilities have the same client ID, the most recent one to connect forcefully disconnects the previous one. When you run more than one MQTT client utility from a telemetry channel, the generated client identifier has a numeric suffix that is incremented every time a new client utility is started.
3. Click **Options** to open the **Connection Options** window. You can start the client utility with a clean session, or configure the last will and testament options.
4. Click **Connect** to establish a connection with the PlainText telemetry channel. A new event entry of Connected is displayed in the **Client history**.
5. Type a topic name in the **Subscription Topic** field. The default topic name is testTopic and this name is used throughout this task.
6. Select the subscription quality of service from the **Request QoS** menu.
7. Click **Subscribe** to subscribe to the topic testTopic. A new event entry of Subscribed is displayed in the **Client history**, along with the topic name, QoS, and the time of subscription.
8. Accept the default topic name, testTopic, in the **Publication Topic** field. In general, ensure that the subscription and publication topics match so that the MQTT client receives messages from the correct topic.
9. Type a message in the **Message** field. The default message test is Test Message.
10. Select the publication quality of service from the **Request QoS** menu.
11. Select **Retained** to forward the most recent retained publication on this topic to new subscribers.
12. Click **Publish** to publish the message on the testTopic topic for interested subscribers. A new event entry of Published is displayed in the **Client history**, along with the topic name, QoS, whether the message is retained, and the time of subscription. On the receiving client utility, a new event entry of Received is displayed in the **Client history**.
13. Select the received message in the **Client history**, then click **View message** to view the full message in the **Message Viewer** window. Alternatively, select the message and press **Enter**, or double-click the received message.

Results

Publishing messages and viewing the messages from topics subscribed to, shows that you have correctly set up your queue manager for Telemetry.

Related concepts

[“MQTT client utility” on page 271](#)

The MQTT client utility is a Java application with which you can explore the features of MQTT, that is, connecting to a queue manager, and publishing and subscribing to topics.

Related tasks

[“Resolving problems if your MQTT client fails to connect” on page 284](#)

If your MQTT client cannot connect to a telemetry channel, there might be a number of reasons.

[“Resolving problems if your MQTT client disconnects unexpectedly” on page 285](#)

Discover what is going wrong when an MQTT client is unexpectedly disconnected from a telemetry channel.

Configuring MQ Telemetry using IBM MQ Explorer

Configure IBM MQ to run the Telemetry feature, using IBM MQ Explorer. Create telemetry objects, and test your telemetry setup using the MQTT client utility.

About this task

Change the MQ Telemetry information by modifying the values specified on a set of configuration attributes that govern the Telemetry feature. The sample configuration provides a basic setup with defined attributes. Change the behavior of the preset telemetry objects by modifying their attributes or properties. For more information about the meaning of each attribute, see [“Telemetry channel properties” on page 286](#) and [“Telemetry channel status attributes” on page 288](#).

Related tasks

[“Creating and configuring a telemetry channel” on page 275](#)

A telemetry channel connects a number of MQTT clients to IBM MQ. Create one or more telemetry channels on a queue manager. Each of these telemetry channels might have different configuration settings, making it easier to manage the clients attached to them.

[“Defining a sample configuration” on page 277](#)

You can use the **Define sample configuration** wizard to reconfigure your queue manager, making it suitable for the MQ Telemetry feature. The sample configuration defines and starts the MQXR service, defines the transmit queue, and creates a sample telemetry channel.

[“Defining the MQXR service” on page 278](#)

The MQXR service is defined when you run the **Define sample configuration** wizard. You can also define the MQXR service manually.

[“Defining the MQXR service manually on Linux” on page 278](#)

[“Defining the MQXR service manually on Windows” on page 279](#)

Creating and configuring a telemetry channel

A telemetry channel connects a number of MQTT clients to IBM MQ. Create one or more telemetry channels on a queue manager. Each of these telemetry channels might have different configuration settings, making it easier to manage the clients attached to them.

Procedure

Create and configure a new telemetry channel by completing the following steps:

1. Right-click the telemetry **Channels** folder and click **New > Telemetry channel**. The **New Telemetry Channel** wizard opens.
2. Type the name of the channel in the **Channel name** field.

The names of telemetry channels are restricted to 20 characters. The characters that can be used in a telemetry channel name, as with all IBM MQ names, are:

- Uppercase A-Z
- Lowercase a-z
- Numerics 0-9
- Period (.)
- Underscore (_)
- Forward slash (/)
- Percent sign (%)

Leading or embedded blanks are not allowed.

3. Type the port number in the **Port number** field. The default port number for a telemetry channel without TLS security is 1883.
4. Optional: If you choose to secure your new telemetry channel using TLS, select **Secure channel using SSL**. The port number changes to 8883, which is the default for a channel secured using TLS.
 - a) Click **Next**.
 - b) Type the name of the SSL/TLS file to use in the **SSL Key File** field.
 - c) Type the password to unlock the key file in the **SSL Passphrase** field.
 - d) Select **Identify client using digital certificate** to force all clients to send their privately signed digital certificate for authentication, or select **Allow anonymous clients** to stop the telemetry channel from authenticating the client using TLS.
5. Click **Next**.
6. Select one of the following options for client authentication:
 - **Don't check client supplied username and password**: Select this option if you want your program to either perform its own authentication, or you do not need any clients authenticated.
 - **Check client supplied username and password (using JAAS)**: Select this option to verify the identity of the client using JASS. Select the name of the JAAS configuration you want to implement from the **JAAS config name** menu.
7. Click **Next**.
8. Select from one of the following menu options:
 - Select **MQTT Client ID** to use the supplied MQTT client ID.
 - Select **Fixed user ID** to disregard whatever user ID is supplied by the client. Type your preferred user ID in the **User ID** field. The default value is Guest on Windows systems, and nobody on Linux systems.
 - Select **Username provided by client** to use the user name supplied in by the client. If no user name is supplied, the client fails to connect to IBM MQ.
9. Click **Next**.
10. Optional: Select **Launch MQTT client utility** to launch the graphical utility to test the MQTT protocol.
11. Review the list of actions to be performed and click **Finish**.

Results

A new telemetry channel is created. View this channel by expanding the **Telemetry** folder and clicking the **Channels** folder.

What to do next

You can now manage your telemetry channel authorities.

For information about how to grant authorities in IBM MQ Explorer, see [“Managing object authorities with an authorization service” on page 165](#).

Related tasks

[“Defining a sample configuration” on page 277](#)

You can use the **Define sample configuration** wizard to reconfigure your queue manager, making it suitable for the MQ Telemetry feature. The sample configuration defines and starts the MQXR service, defines the transmit queue, and creates a sample telemetry channel.

[“Defining the MQXR service” on page 278](#)

The MQXR service is defined when you run the **Define sample configuration** wizard. You can also define the MQXR service manually.

Defining a sample configuration

You can use the **Define sample configuration** wizard to reconfigure your queue manager, making it suitable for the MQ Telemetry feature. The sample configuration defines and starts the MQXR service, defines the transmit queue, and creates a sample telemetry channel.

Before you begin

Before you run the **Define sample configuration** wizard:

- You must install the IBM MQ Telemetry feature on the computer.
- You must be willing to let the wizard reconfigure your queue manager. If you are unsure of the implications, create a new queue manager solely for this purpose, or see [“Implications of running the sample configuration” on page 283](#).

About this task

By using the sample configuration to get started, you can set up a basic configuration on your computer so that you can explore the telemetry features. You cannot run the sample configuration twice on the same queue manager unless you remove one or more of the IBM MQ object definitions created. When an object definition created by the sample configuration is deleted, running the wizard again only re-creates that missing object.

Procedure

1. From the Telemetry welcome page, click **Define sample configuration**. The **Define sample configuration** wizard opens.
2. Review the list of actions that will occur on completion of this wizard, and click **Finish**.

Results

The **Define sample configuration** wizard performs the following actions and creates the appropriate resources:

- Defines and starts the MQXR service.
- Defines the default transmit queue.
- Allows Guest on Windows systems, and nobody on Linux systems, to send messages to clients connected to the MQTT listener.
- Allows Guest on Windows systems, and nobody on Linux systems, to both publish on and subscribe to any topic.
- Defines a sample telemetry channel.

Also, the **Define sample configuration** link on the Telemetry welcome page is replaced by **The sample configuration has been set up for this queue manager**. This is the first form of visual verification that the sample configuration was set up properly.

What to do next

You can view the sample telemetry channel that the wizard created by expanding the **Telemetry** folder and clicking the **Channels** folder.

If you remove one of the definitions created by the sample configuration wizard, you can run the wizard again. The wizard creates the same resource that you deleted and informs you in the summary section.

Defining the MQXR service

The MQXR service is defined when you run the **Define sample configuration** wizard. You can also define the MQXR service manually.

About this task

Running the **Define sample configuration** wizard creates some IBM MQ objects and resources. One of these objects is the MQXR service. For more information about running the **Define sample configuration** wizard, see [“Defining a sample configuration” on page 277](#).

You can also define the MQXR service manually by performing a list of steps. For more information, see [“Defining the MQXR service manually on Windows” on page 279](#) and [“Defining the MQXR service manually on Linux” on page 278](#).

Results

The creation of an expansible **Telemetry** folder node indicates the successful definition of the MQXR service.

Related tasks

[“Telemetry node does not appear” on page 285](#)

Find out what to look for if the Telemetry node does not appear.

Defining the MQXR service manually on Linux

You can define the MQXR service manually by using IBM MQ Explorer. A queue manager can have only one instance of the MQXR service defined.

Before you begin

- Install the MQ Telemetry feature.

Procedure

1. In the Navigator view, right-click the **Services** folder.
2. Click **New > Service** to open the **New Service Definition** wizard.
3. In the **Name** field, type `SYSTEM.MQXR.SERVICE` and click **Next**.
4. In the **Description** field type a description of the service (for example, `Manages clients using MQXR protocols such as MQTT`).
5. Select an option from the **Service control** menu.
6. In the **Start command** field, type `+MQ_INSTALL_PATH+/mqxr/bin/runMQXRService.sh`
7. In the **Start args** field, type `-m +QMNAME+ -d "+MQ_Q_MGR_DATA_PATH+" -g "+MQ_DATA_PATH+"`
8. In the **Stop command** field, type `+MQ_INSTALL_PATH+/mqxr/bin/endMQXRService.sh`
9. In the **Stop args** field, type `-m +QMNAME+`
10. In the **StdOut** field, type `+MQ_Q_MGR_DATA_PATH+/mqxr.stdout`
11. In the **StdErr** field, type `+MQ_Q_MGR_DATA_PATH+/mqxr.stderr`
12. Select **Server** from the **Service type** menu.
13. Click **Finish**.

Results

The MQXR service is created.

To view the MQXR service in the Navigator view click the **Services** folder. Ensure the **Show System Objects** option is selected, and navigate to the service.

In this task, the service is called SYSTEM.MQXR.SERVICE.

Related tasks

[“Defining the MQXR service manually on Windows” on page 279](#)

Defining the MQXR service manually on Windows

You can define the MQXR service manually by using IBM MQ Explorer. A queue manager can have only one instance of the MQXR service defined.

Before you begin

- Install the MQ Telemetry feature.

Procedure

1. In the Navigator view, right-click the **Services** folder.
2. Click **New > Service** to open the **New Service Definition** wizard.
3. In the **Name** field, type `SYSTEM.MQXR.SERVICE` and click **Next**.
4. In the **Description** field, type a description of the service (for example, `Manages clients using MQXR protocols such as MQTT`).
5. Select an option from the **Service control** menu.
6. In the **Start command** field, type `+MQ_INSTALL_PATH+\mqxr\bin\runMQXRService.bat`
7. In the **Start args** field, type `-m +QMNAME+ -d "+MQ_Q_MGR_DATA_PATH+\" -g "+MQ_DATA_PATH+\"`
8. In the **Stop command** field, type `+MQ_INSTALL_PATH+\mqxr\bin\endMQXRService.bat`
9. In the **Stop args** field, type `-m +QMNAME+`
10. In the **StdOut** field, type `+MQ_Q_MGR_DATA_PATH+\mqxr.stdout`
11. In the **StdErr** field, type `+MQ_Q_MGR_DATA_PATH+\mqxr.stderr`
12. Select **Server** from the **Service type** menu.
13. Click **Finish**.

Results

The MQXR service is created.

To view the MQXR service in the Navigator view click the **Services** folder. Ensure the **Show System Objects** option is selected, and navigate to the service.

In this task, the service is called SYSTEM.MQXR.SERVICE.

Related tasks

[“Defining the MQXR service manually on Linux” on page 278](#)

Administering MQ Telemetry using IBM MQ Explorer

MQ Telemetry can be administered by using IBM MQ Explorer. You can control the MQXR service, and monitor the MQTT clients that are connected to IBM MQ.

About this task

For information about client authorization, authenticating a telemetry channel using TLS, and JAAS configurations, see [Administering IBM MQ Telemetry](#) in the IBM MQ online product documentation.

Related tasks

[“Starting and stopping the MQXR service” on page 280](#)

Before you can start or stop the MQXR service, the queue manager must be running.

[“Starting and stopping a telemetry channel” on page 280](#)

[“Viewing the status of a telemetry channel” on page 281](#)

[“Filtering Telemetry objects” on page 282](#)

If you are viewing several defined telemetry objects in the **Content** view, you might need a way to narrow the search scope of these objects. Do this by using filters.

Starting and stopping the MQXR service

Before you can start or stop the MQXR service, the queue manager must be running.

About this task

When you start the MQXR service on a queue manager, it listens on the receiving ends of the telemetry channels on that queue manager for incoming messages from clients.

Stopping the MQXR service has the following implications:

- The MQXR service does not listen for client connections.
- The **Telemetry** folder cannot be expanded. This means that you cannot:
 - Create or view telemetry channels
 - View client connections
 - Send messages to clients
 - Receive messages from clients

You can modify the properties of an MQXR service in a similar way to a standard IBM MQ service. Right-click the service name, and click **Properties**.

In the **Properties** window, you can configure the MQXR service to start and stop with the queue manager, or start and stop manually, by selecting the appropriate option from the **Service control** menu.

Procedure

Use the following steps to start or stop the MQXR service:

1. In the **Navigator** view, click the **Services** folder.
2. Ensure that **Show System Objects** is selected.
3. In the **Content** view, right-click the MQXR service name (SYSTEM.MQXR.SERVICE), and click **Start** or **Stop**.
4. Click **Yes** on the confirmation dialog.

Results

The MQXR service starts or stops depending on what action you selected.

Related tasks

[“Defining the MQXR service” on page 278](#)

The MQXR service is defined when you run the **Define sample configuration** wizard. You can also define the MQXR service manually.

Starting and stopping a telemetry channel

A telemetry channel is automatically started when it is created. It stops when the queue manager or MQXR service is stopped. You can also start and stop a telemetry channel manually.

A telemetry channel also stops when it is purged. Purging a telemetry channel disconnects all the MQTT clients connect to it, cleans up the state of the MQTT clients, and stops the telemetry channel. Cleaning

the state of a client involves deleting all the pending publications, and removing all the subscriptions from the client.

Before you begin

Ensure that the MQXR service is defined and running.

Procedure

Start or stop a telemetry channel manually, by performing the following steps:

1. In the **Navigator** view, expand the **Telemetry** folder.
2. Click **Channels**.
3. In the **Content** view, select the telemetry channel you want to start or stop.
4. Right-click the selected telemetry channel, then click **Start** or **Stop**.

Results

The telemetry channel starts or stops depending on what action you performed.

Note: To purge a telemetry channel, right-click the selected channel and click **Purge**.

Related tasks

[“Creating and configuring a telemetry channel” on page 275](#)

A telemetry channel connects a number of MQTT clients to IBM MQ. Create one or more telemetry channels on a queue manager. Each of these telemetry channels might have different configuration settings, making it easier to manage the clients attached to them.

[“Starting and stopping the MQXR service” on page 280](#)

Before you can start or stop the MQXR service, the queue manager must be running.

Viewing the status of a telemetry channel

The status of a running telemetry channel provides information about the clients connected to it.

A telemetry channel status object can be purged. Purging an MQTT client connection disconnects the client from a telemetry channel and cleans up the state of the client. Cleaning the state of an MQTT client involves deleting all pending publications, and removing all subscriptions from that client.

Procedure

To view the status of a telemetry channel, perform the following steps:

1. In the **Navigator** view, expand the **Telemetry** folder, then click the **Channels** folder. Your telemetry channel definitions are displayed in the **Content** view.
2. Right-click the appropriate telemetry channel, then click **Status**. A new **Content** view opens in a separate window displaying the client connections on that telemetry channel.

Viewing all client connections to telemetry channels

It is possible to view all the client connections made to all telemetry channels on a queue manager. To do this, expand the **Telemetry** folder, then click the **Channel status** folder in the **Navigator** view.

All client connections to every telemetry channel on that queue manager are displayed in the **Content** view. You can reorder the display of objects by clicking the relevant column name to sort the results. Alternatively, you can use filtering.

By default, IBM MQ Explorer displays only the first 500 client connections. For more information about how to configure the maximum number of connections displayed at any given time, see [“Advanced option” on page 270](#).

Note: To purge a telemetry channel status object, right-click the selected object and click **Purge**.

Related tasks

[“Filtering Telemetry objects” on page 282](#)

If you are viewing several defined telemetry objects in the **Content** view, you might need a way to narrow the search scope of these objects. Do this by using filters.

Filtering Telemetry objects

If you are viewing several defined telemetry objects in the **Content** view, you might need a way to narrow the search scope of these objects. Do this by using filters.

Before you begin

- Install the MQ Telemetry feature.
- Configure your queue manager for the MQ Telemetry feature. For more information, see [“Configuring MQ Telemetry using IBM MQ Explorer” on page 275](#).

About this task

The **Telemetry channel status** view has a filtering option. Within Telemetry, filtering is best used when viewing the client connections in the **Telemetry Channel Status Content** view. Multiple telemetry channels might have multiple clients attached to each of them. All of these connections are displayed in the **Channel Status** folder. For example, you might want to display MQTT clients with client IDs like `ibm_client`. This returns clients with client IDs such as `ibm_client1`, `ibm_client2`, and `ibm_client3`.

You can also filter telemetry channels using the same filtering steps. For this task, you are filtering in the **Telemetry Channel Status** content view.

Procedure

To filter telemetry objects, perform the following steps:

1. Assuming that you have installed and set up your queue manager for Telemetry, click the **Channel Status** folder.
2. In the **Telemetry Channel status Content** view, click the arrow next to the **Filter** name.
 - To select a filtering option from a list of defined filters, click **Select Filter**. The default filter in the **Channel Status Content** view is **Standard for Telemetry Channel Status**.
 - To change the options for the current filter, click **Edit current filter**.
 - To add, copy, or edit filters, click **Manage filters**.
 - a) To add a filter, in the **Manage filters** window, click **Add**.
 - b) Type a meaningful name in the **Filter Name** field. For example, type `Clients belonging to my IBM channel`.
 - c) Set the condition to apply to the telemetry channels. For example, `Channel name like IBM.CHANNEL`.
 - d) To add another rule, select **AND**.
 - e) Click **Select** to change the attribute to filter on.
 - f) Type an appropriate rule, then click **OK**.
3. Select the filter name that you want to apply to that content view and click **OK**.

Results

The filter is applied and your objects are filtered based on the criteria set in the filtering option.

Troubleshooting MQ Telemetry using IBM MQ Explorer

Help is provided for some of the problems that might occur when you use IBM MQ Explorer to administer telemetry.

About this task

When you install the Telemetry feature, you might want to run the **Define sample configuration** wizard to set up a basic configuration of telemetry. See [“Defining a sample configuration” on page 277](#).

You can verify and test your basic setup by using the MQTT client utility to publish, and subscribe to, messages. For more information about testing your sample configuration, see [“Using the MQTT client utility” on page 274](#).

Related tasks

[“Resolving problems if your MQTT client fails to connect” on page 284](#)

If your MQTT client cannot connect to a telemetry channel, there might be a number of reasons.

[“Resolving problems if your MQTT client disconnects unexpectedly” on page 285](#)

Discover what is going wrong when an MQTT client is unexpectedly disconnected from a telemetry channel.

[“Telemetry node does not appear” on page 285](#)

Find out what to look for if the Telemetry node does not appear.

[“Resolving problems with a telemetry channel” on page 285](#)

If a telemetry channel fails to start, stops unexpectedly, or drops client connections, there are a few things to consider to diagnose the problem.

Related reference

[“Implications of running the sample configuration” on page 283](#)

When you run the **Define sample configuration** wizard, IBM MQ objects are defined. Some of these objects alter the behavior of the queue manager, and you should be aware of how these objects affect the queue manager and its communication links.

Implications of running the sample configuration

When you run the **Define sample configuration** wizard, IBM MQ objects are defined. Some of these objects alter the behavior of the queue manager, and you should be aware of how these objects affect the queue manager and its communication links.

Running the **Define sample configuration** wizard sets the default transmit queue of the queue manager to `SYSTEM.MQTT.TRANSMIT.QUEUE`, which takes precedence over an existing default transmit queue, if one existed on that queue manager.

Defining the default transmit queue to be the MQTT transmit queue, enables IBM MQ applications to send point-to-point messages to MQTT clients without the need to create a separate queue manager alias for every client. Messages destined for MQTT clients are routed through the MQTT transmit queue on the queue manager, to the MQTT client with a client identifier that matches the queue manager name that the message is sent to. IBM MQ queue managers perceive MQTT clients as though they were remote queue managers.

If you had previously used an IBM MQ default transmit queue to route messages to other queue managers, you must explicitly create alternative routes (for example, by defining queue manager aliases) before running the sample configuration or manually configuring your queue manager to enable the Telemetry feature.

Running the sample configuration causes MQTT clients to access IBM MQ resources with user name `Guest` on Windows, and `nobody` on Linux.

Resolving problems if your MQTT client fails to connect

If your MQTT client cannot connect to a telemetry channel, there might be a number of reasons.

Procedure

Consider the following reasons to diagnose the problem with an MQTT client failing to connect:

- Check that the queue manager and telemetry (MQXR) service are running.
Start the queue manager. By default, the MQXR service should start with the queue manager. If you configured the MQXR service control to start manually, you might have to start the service from the **Services** folder. For more information about starting the MQXR service, see [“Starting and stopping the MQXR service” on page 280](#).
- Check that the telemetry channel and telemetry (MQXR) service are defined and running correctly.
You can manually define the MQXR service and set the default transmit queue of the queue manager to `SYSTEM.MQTT.TRANSMIT.QUEUE`, which takes precedence over an existing default transmit queue. This makes the queue manager suitable for Telemetry. Alternatively, you might want to consider running the **Define sample configuration** wizard from the Telemetry welcome page, if you have not done so already.
- Have you written your own client?
If so, did you write your client application with the MQTT v3 protocol and not the v5 protocol? Try to isolate the problem by running the MQTT client utility.
- Do you have a valid client identifier name?
When connecting to IBM MQ, the MQTT client identifier should be less than 23 characters, and contain only alphabetic characters, numeric characters, and the period (.), forward slash (/), underscore (_), and percent sign (%).
- Did you connect your MQTT client and exhaust the **MQTT keep alive** interval?
The keep alive attribute is the interval in milliseconds after which, the MQTT client is disconnected due to inactivity. If the MQXR service does not receive any communication from the client within the keep alive interval, it disconnects from the client.
- Is a large number of MQTT clients trying to connect to a telemetry channel at the same time?
Every telemetry channel has a **backlog** attribute. This is the number of concurrent connection requests that the telemetry channel supports. Ensure that the value is not set to a number that is less than the number of MQTT clients trying to connect.
- Check that the TCP/IP connection is still active.

Related tasks

[“Defining a sample configuration” on page 277](#)

You can use the **Define sample configuration** wizard to reconfigure your queue manager, making it suitable for the MQ Telemetry feature. The sample configuration defines and starts the MQXR service, defines the transmit queue, and creates a sample telemetry channel.

[“Defining the MQXR service” on page 278](#)

The MQXR service is defined when you run the **Define sample configuration** wizard. You can also define the MQXR service manually.

Related reference

[“Telemetry channel properties” on page 286](#)

Each telemetry channel attribute has a brief description which you must understand before you can configure the channel. MQ Telemetry only supports the TCP/IP protocol.

[“Telemetry channel status attributes” on page 288](#)

As with IBM MQ, you can view the status of a telemetry channel. For each attribute, there is a brief description of what information the attribute is used for. All of the telemetry channel status attributes are read-only.

Resolving problems if your MQTT client disconnects unexpectedly

Discover what is going wrong when an MQTT client is unexpectedly disconnected from a telemetry channel.

Procedure

If your MQTT client connects successfully and later disconnects with no apparent reason, consider the following reasons to diagnose the problem:

- The queue manager, MQXR service, or telemetry channel is not running.
Start the queue manager, MQXR service, or telemetry channel. Try reconnecting the MQTT client, and check that this solution rectifies the problem.
- Another client is started and connects with the same client ID.
In this case, IBM MQ accepts the connection from the second MQTT client and forcefully disconnects the first MQTT client.
- The MQTT client accesses a topic that it is not authorized to, either for publishing or subscribing.
IBM MQ disconnects the MQTT client.
- The TCP/IP connection is no longer active.
Diagnose and fix the problem with your TCP/IP connection, and try reconnecting the MQTT client.

Telemetry node does not appear

Find out what to look for if the Telemetry node does not appear.

Procedure

- Have you installed MQ Telemetry?
Check that you have all the prerequisites, and that you have installed Telemetry. See *Installation* under *Telemetry* in the IBM MQ product documentation.

Resolving problems with a telemetry channel

If a telemetry channel fails to start, stops unexpectedly, or drops client connections, there are a few things to consider to diagnose the problem.

Procedure

- Your telemetry channel fails to start.
Refresh the **Telemetry Channels Content** view, and ensure that the channel is not currently running.
Check that the port number of the telemetry channel is not in use by another application.
- A telemetry channel stops unexpectedly.
Ensure that the telemetry (MQXR) service is still running.
- The telemetry channel drops MQTT client connections.
For more information about MQTT clients being dropped unexpectedly, see [“Resolving problems if your MQTT client disconnects unexpectedly” on page 285](#).
- You cannot view the status of a telemetry channel.
Check that the telemetry channel in question is running.

Ensure that the MQTT clients are connected to the correct telemetry channel. Check the port number and host name on the client matches that of the telemetry channel.

If you have set up your own filter in the telemetry **Channel Status Window**, revert to the default of **Standard for Telemetry Channel Status** and check that the required MQTT clients are displayed.

- You cannot run the MQTT client utility from a telemetry channel.

If you choose to perform authentication using TLS or JAAS, you cannot launch the MQTT client utility from that telemetry channel. This is because the MQTT client utility does not support TLS or JAAS authentication. However, you can write your own MQTT v3 client application to support JAAS or TLS authentication.

- The **Telemetry Channels** folder does not display any channels, or the channel you created.

Check that you have successfully created a telemetry channel using either the **Define sample configuration** wizard (which creates the PlainText channel), or the **New Telemetry Channel** wizard (which creates a channel according to your specifications).

Check that the filtering option is set to the telemetry channel default of **Standard for Telemetry Channels**.

Related tasks

[“Filtering Telemetry objects” on page 282](#)

If you are viewing several defined telemetry objects in the **Content** view, you might need a way to narrow the search scope of these objects. Do this by using filters.

MQ Telemetry Reference

Use the reference information in this section to accomplish tasks associated with the use of Telemetry.

Related reference

[“Telemetry channel properties” on page 286](#)

Each telemetry channel attribute has a brief description which you must understand before you can configure the channel. MQ Telemetry only supports the TCP/IP protocol.

[“Telemetry channel status attributes” on page 288](#)

As with IBM MQ, you can view the status of a telemetry channel. For each attribute, there is a brief description of what information the attribute is used for. All of the telemetry channel status attributes are read-only.

Telemetry channel properties

Each telemetry channel attribute has a brief description which you must understand before you can configure the channel. MQ Telemetry only supports the TCP/IP protocol.

Right-click any telemetry object, then click **Properties** to view and edit the telemetry channel's properties.

The following table lists the attributes that you can set on the **Telemetry channel properties** dialog ([Note 1](#)).

Attribute	Meaning
Channel name	Read-only. This is the name of the telemetry channel definition.
Channel type	Read-only. This is the type of channel, in this case, MQTT.
Overall channel status	Read-only. This is the current status of the telemetry channel.
Xmit protocol	Read-only. The transmission protocol of the channel. Only TCP/IP is supported.

Attribute	Meaning
Port	This is the port number that the MQXR service accepts client connections on. The default port number for a telemetry channel is 1883; and the default port number for a telemetry channel secured using TLS is 8883.
Local address (optional)	Type the IP address that the telemetry channel listens on. Use this option when a server has multiple IP addresses.
Backlog (optional)	The number of outstanding connection requests that the telemetry channel can support at any one time. When the backlog limit is reached, any further clients trying to connect will be refused connection until the current backlog is processed. The value is in the range 0 - 999999999. The default value is 4096.
MCA user ID (optional) See Note 2	The user ID for the message channel agent. It is the user identifier (up to 12 characters) to be used by the MCA for authorization to access IBM MQ resources. If this property is specified, the user name supplied by the client is not used for IBM MQ authorization.
Use client ID (optional) See Note 2	Decide whether you want to use the MQTT client ID for the new connection as the IBM MQ user ID for that connection. If this property is specified, the user name supplied by the client is ignored.
SSL CipherSuite (optional)	If you choose to use this property, the CipherSuite must be available at the client end of the telemetry channel. Leaving this option blank makes both ends of the telemetry channel negotiate a CipherSuite that they both understand.
SSL Authentication (optional)	Determines whether the client is treated anonymously. SSL authentication defines whether the telemetry channel must receive and authenticate a TLS certificate from a client.
SSL Key repository (optional)	The store for digital certificates and their associated private keys. If you do not specify a key file, TLS is not used.
SSL Passphrase (optional)	The password for the key repository. If no passphrase is entered, then unencrypted connections must be used.
JAAS config file (read-only)	The file path of the JAAS configuration.
JAAS config name (optional)	The name of the configuration in the <code>jaas.config</code> file that you want to implement.

Note:

1. When you edit the attributes of a telemetry channel, you must restart the channel for the changes to apply.
2. Do not specify both **MCA user ID** and **Use client ID** properties. If you specify both, the telemetry channel will fail when it tries to start.

If neither the **MCA user ID** and **Use client ID** properties are set, the user name and password from the client are used and the user name is authenticated by JAAS using the password.

Related tasks

[“Configuring MQ Telemetry using IBM MQ Explorer” on page 275](#)

Configure IBM MQ to run the Telemetry feature, using IBM MQ Explorer. Create telemetry objects, and test your telemetry setup using the MQTT client utility.

[“Administering MQ Telemetry using IBM MQ Explorer” on page 279](#)

MQ Telemetry can be administered by using IBM MQ Explorer. You can control the MQXR service, and monitor the MQTT clients that are connected to IBM MQ.

Telemetry channel status attributes

As with IBM MQ, you can view the status of a telemetry channel. For each attribute, there is a brief description of what information the attribute is used for. All of the telemetry channel status attributes are read-only.

The following table provides the descriptions of telemetry channels status attributes:

Attribute	Meaning
Channel name	The name of the telemetry channel definition.
Client ID	The identifier of the client.
Status	The status of the client, which can be Running or Disconnected.
Indoubt in	The number of indoubt inbound messages to the server. Indoubt inbound messages are messages that have been received by the server but have not completed the acknowledgments with the client.
Indoubt out	The number of indoubt outbound messages from the server. Indoubt outbound messages are messages that have been sent by the server but have not had acknowledgments of receipt from the client.
Connection name	The name of the remote connection. The connection name is always an IP address, or it could be localhost (127.0.0.1).
MQTT keep alive	The interval in milliseconds after which the client is disconnected because of inactivity. If the MQXR service does not receive any communication from the client within the keep alive interval, it disconnects from the client. This interval is calculated based on the MQTT keep alive time sent by the client when it connects.
MCA user ID	The message channel agent user identification string. It is the user identifier (1-12 characters) to be used by the MCA for authorization to access IBM MQ resources. If this property is specified, the user name supplied by the client is not used for IBM MQ authorization.
Messages sent	The number of messages sent by the telemetry channel to the client since the most recent client connection session.
Messages received	The number of messages received by the telemetry channel from the client since the most recent client connection session.
Last message time	The time the last message was sent or received.
Channel start time	The time the telemetry channel was started.
Pending out	The number of outbound pending messages on the telemetry channel waiting to be sent to the MQTT client.
Channel start date	The date the telemetry channel was started.

Related tasks

[“Viewing the status of a telemetry channel” on page 281](#)

[“Filtering Telemetry objects” on page 282](#)

If you are viewing several defined telemetry objects in the **Content** view, you might need a way to narrow the search scope of these objects. Do this by using filters.

Multi IBM MQ Tutorials

These tutorials show how to perform basic tasks such as creating a queue manager, creating a queue, creating a channel, putting a message onto a queue, and getting a message from a queue. The tutorials are relevant only for Multiplatforms.

Each tutorial is divided into several subtasks. You can perform each task by using either of the following interfaces:

- The IBM MQ Explorer graphical interface.
- The IBM MQ Script Commands (MQSC) command line interface.

See the [MQSC reference](#) section for more information on these commands.

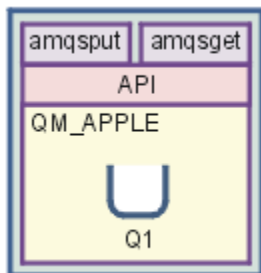
The first tutorial shows you how to set up a simple local stand-alone installation that has no communication links with any other installations of IBM MQ. Each of the subsequent tutorials then builds upon IBM MQ objects that have been set up during previous tutorials. Therefore, it is recommended that these tutorials are completed in order.

The tutorials are designed to get you started with IBM MQ, and do not cover the more complex messaging scenarios.

Tutorial 1: Sending a message to a local queue

A tutorial to set up a queue manager, create a queue, put a test message onto the queue, and verify the receipt of the message.

About this task



This tutorial shows you how to set up a queue manager QM_APPLE and a queue Q1 on a local stand-alone installation that has no communication links with any other installations of IBM MQ. When the objects have been defined, there are multiple tools that may be used to test the setup. The first task is to put a test message. This task can be completed by using the IBM MQ Explorer, or the distributed platform amqsput program. The second task is to verify that the message was added to the queue. This task can be completed by using the IBM MQ Explorer, or the distributed platform amqsget program.

When you have completed Tutorial 1, you should have a basic understanding of how IBM MQ messaging works in a simple messaging topology that has a queue manager with local queues.

Creating the queue manager

Before you begin

Before creating a queue manager, you must ensure that IBM MQ is correctly installed.

About this task

This topic shows how to create a queue manager called QM_APPLE.

You can perform this task using either of the following interfaces:

- [The IBM MQ Explorer graphical interface](#)
- [IBM MQ Script Command \(MQSC\) command-line interface](#)

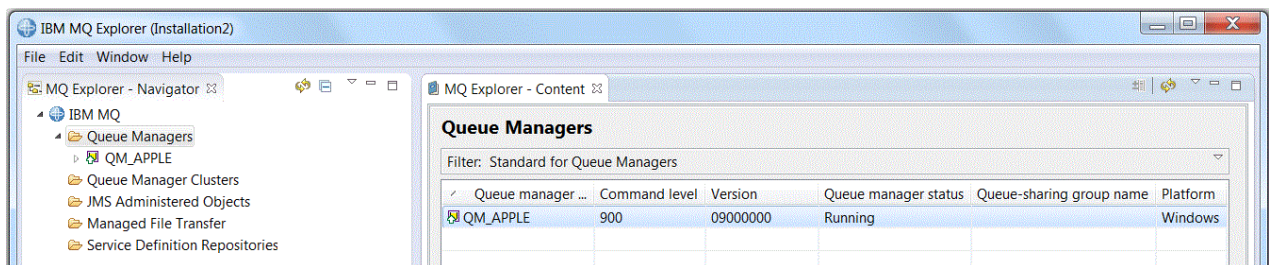
Creating the queue manager using IBM MQ Explorer

Procedure

1. Start IBM MQ Explorer.
2. In the **Navigator view**, right-click the **Queue Managers** folder, then click **New > Queue Manager**.
The **Create Queue Manager** wizard opens.
3. In the **Queue Manager name** field, type QM_APPLE.
4. Click **Next** twice.
5. Ensure that **Automatic** is selected from the **Select type of queue manager startup** option.
6. Click **Next**.
7. Ensure that the **Create listener configured for TCP/IP** check box is selected.
8. If the **Finish** button is not available, type another port number in the **Listen on port number** field.
If the current value is 1414, try using a different port number, for example: 1415 or 1416. If the default port number of 1414 is not used at this stage, make a note of the port number used because you will need it in later stages of this tutorial when QM_APPLE serves as a receiving queue manager.
9. Click **Finish**.

Results

An icon representing this queue manager is displayed in the **Queue Managers** folder in the **Navigator view** of IBM MQ Explorer, and the queue manager automatically starts running after you create it, as shown in the following screen capture:



Creating the queue manager using MQSC

About this task

Open a command prompt, and follow these steps:

Procedure

1. Create a queue manager called QM_APPLE by typing the command:

```
crtmqm QM_APPLE
```

Messages tell you that the queue has been created and that the default IBM MQ objects have been created.

2. Start this queue manager by typing the command:

```
strmqm
```

A message tells you when the queue manager has started.

Results

You have now created a queue manager with the name QM_APPLE.

Creating the local queue

Before you begin

Before creating a local queue on the queue manager, you must already have completed the task: [Creating the queue manager](#).

About this task

This topic shows how to create a local queue called Q1 on the queue manager called QM_APPLE.

You can perform this task using either of the following interfaces:

- The IBM MQ Explorer graphical interface
- IBM MQ Script Command (MQSC) command-line interface

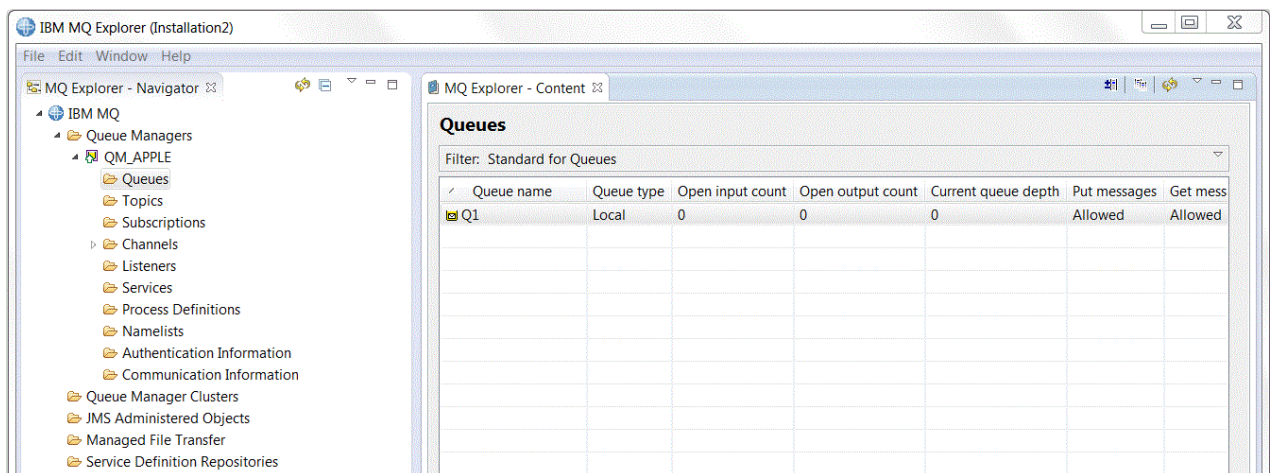
Creating the local queue using IBM MQ Explorer

Procedure

1. In the **Navigator view**, expand the **Queue Managers** folder.
2. Expand queue manager QM_APPLE.
3. Right-click the **Queues** folder, then click **New > Local Queue**.
The **New Local Queue** wizard opens.
4. In the **Name** field, type Q1
5. Click **Finish**.

Results

The new queue Q1, is displayed in the **Content view**, as displayed in the following screen capture:



If the queue is not displayed in the **Content view**, click Refresh  in the **Content view**.

About this task

Open a command prompt and follow these steps:

Procedure

1. Enable MQSC commands by typing the command:

```
runmqsc QM_APPLE
```

2. Type the following command:

```
define qlocal (Q1)
```

Messages tell you that the queue has been created and that the default IBM MQ objects have been created.

3. Stop MQSC by typing the command:

```
end
```

Results

You have now created a local queue called Q1.

Putting a test message on the local queue

Before you begin

Before putting a test message on the queue, you must already have completed the following tasks in this tutorial:

- [Creating the local queue manager](#)
- [Creating the local queue](#)

About this task

This topic shows how to put a test message on the local queue Q1.

You can perform this task using either of the following interfaces:

- [The IBM MQ Explorer graphical interface](#)
- [IBM MQ Script Command \(MQSC\) command-line interface](#)

Putting a test message on the queue using IBM MQ Explorer

Procedure

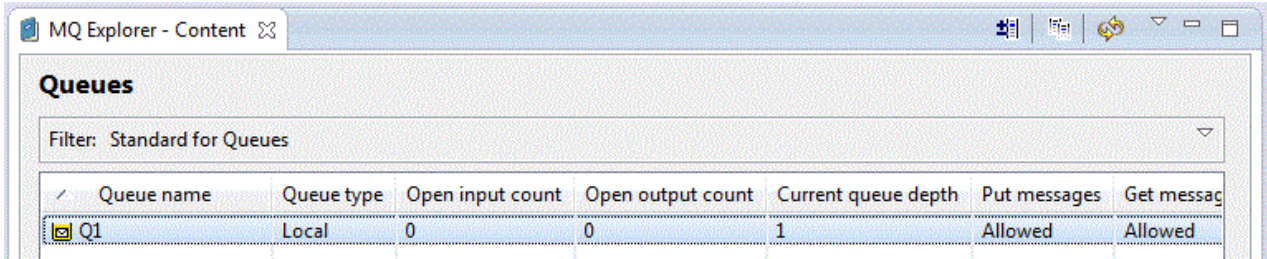
1. In the **Navigator view**, expand the **Queue Managers** folder.
2. Expand queue manager QM_APPLE, which you created.
3. Click the **Queues** folder.
The queues of the queue manager are listed in the **Content view**.
4. In the **Content view**, right-click the local queue Q1, then click **Put Test Message**.
The **Put test message** dialog opens.
5. In the **Message data** field, type some text, for example `this is a test message`, then click **Put message**.

The **Message data** field is cleared and the message is put on the queue.

6. Click **Close**.

Results

In the **Content view**, notice that Q1's **Current queue depth** value is now 1, as shown in the following screen capture:



The screenshot shows the 'MQ Explorer - Content' window. It has a 'Queues' tab selected. Below the tab is a filter dropdown set to 'Standard for Queues'. A table lists the queues. The first row is for queue 'Q1', which is a 'Local' queue. The 'Open input count' and 'Open output count' are both 0. The 'Current queue depth' is 1. The 'Put messages' and 'Get messages' columns show 'Allowed'.

Queue name	Queue type	Open input count	Open output count	Current queue depth	Put messages	Get messages
Q1	Local	0	0	1	Allowed	Allowed

If the **Current queue depth** column is not visible, you might need to scroll sideways in the Content View.

Putting a test message on the queue using amqspout

About this task

The **amqspout** sample program is used to put a message on the queue that you created.

On Windows the sample programs are installed by default with IBM MQ Server or Client. On Linux, the samples programs RPM need to be installed.

Open a command prompt and follow these steps:

Procedure

1. Start the **amqspout** sample program as follows:

- ▶ **Linux** On Linux, change to the `MQ_INSTALLATION_PATH/samp/bin` directory, where `MQ_INSTALLATION_PATH` represents the high-level directory in which IBM MQ is installed. Type the command:

```
./amqspout Q1 QM_APPLE
```

- ▶ **Windows** On Windows, type the command:

```
amqspout Q1 QM_APPLE
```

The following messages are displayed:

```
Sample AMQSPUT0 start
```

```
target queue is Q1
```

2. Type some message text on one or more lines, then press **Enter** twice.

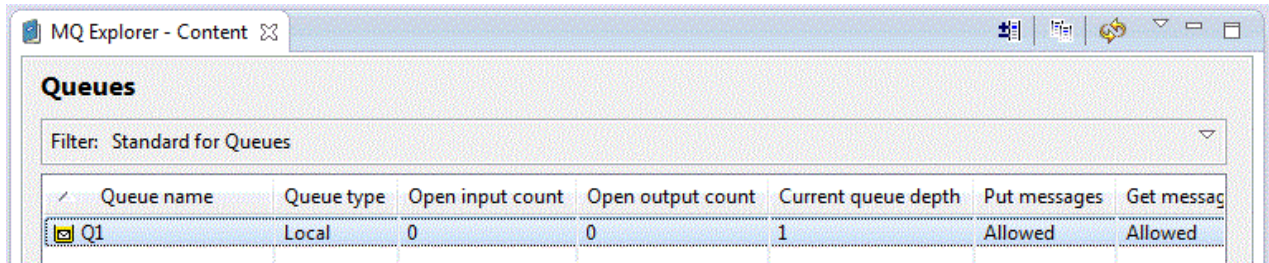
The following message is displayed:

```
Sample AMQSPUT0 end
```

Results

You have now created a test message and put it onto the local queue.

In the IBM MQ Explorer **Content view**, notice that Q1's **Current queue depth** value is now 1, as shown in the following screen capture:



Queue name	Queue type	Open input count	Open output count	Current queue depth	Put messages	Get messages
Q1	Local	0	0	1	Allowed	Allowed

If the **Current queue depth** column is not visible, you might need to scroll sideways in the Content View.

Verifying that the test message was sent

Before you begin

Before getting the test message from the local queue, you must already have completed the following tasks in this tutorial:

- [Creating the local queue manager](#)
- [Creating the local queue](#)
- [Putting a test message on the local queue](#)

About this task

This topic shows how to verify that the test message was sent.

You can perform this task using either of the following interfaces:

- [The IBM MQ Explorer graphical interface](#)
- [IBM MQ Script Command \(MQSC\) command-line interface](#)

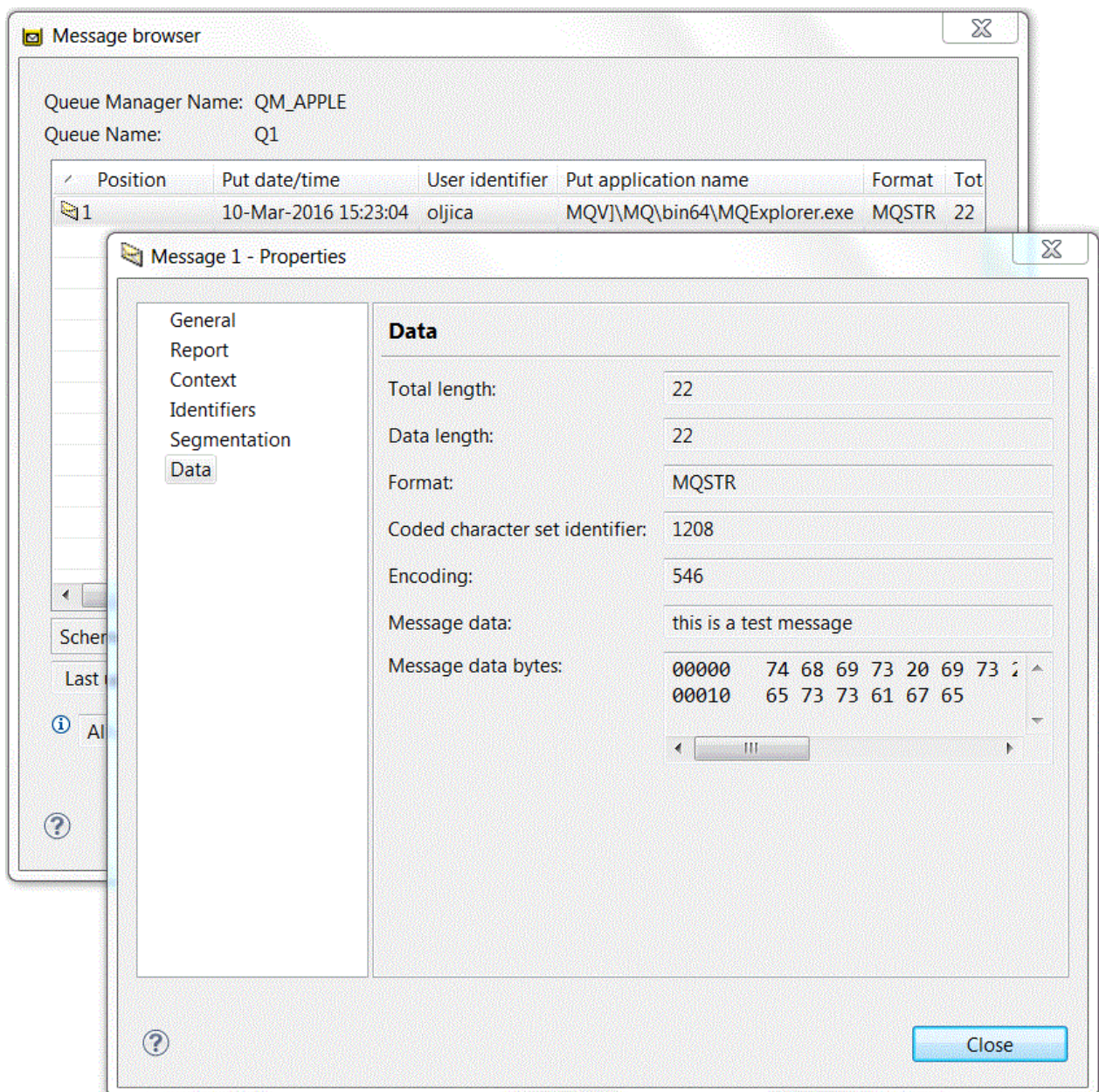
Verifying that the test message was sent using IBM MQ Explorer

Procedure

1. In the **Navigator view**, expand the **Queue Managers** folder, then expand QM_APPLE.
2. Click the **Queues** folder.
3. In the **Content view**, right-click Q1, then click **Browse Messages**.
The **Message browser** opens to show the list of the messages that are currently on Q1.
4. Double-click the last message to open its properties dialog.

Results

On the **Data** page of the properties dialog, the **Message data** field displays the content of the message in human-readable form, as shown in the following screen capture:



Verifying that the test message was sent

About this task

The **amqsget** sample program is used to get the message back from the queue.

Open a command prompt and follow these steps:

Procedure

Start the **amqsget** sample program:

- **Windows** On Windows, type the following command:

```
amqsget Q1 QM_APPLE
```

- Linux On Linux, change to the `MQ_INSTALLATION_PATH/samp/bin` directory, where `MQ_INSTALLATION_PATH` represents the high-level directory in which IBM MQ is installed. Type the following command:

```
./amqsget Q1 QM_APPLE
```

Results

The sample program starts, and your message is displayed along with any other messages on this queue. After a pause of 15 seconds, the sample ends and the command prompt is displayed again.

You have now completed this tutorial.

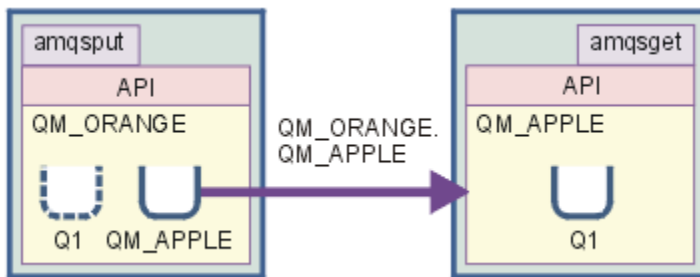
Tutorial 2: Sending a message to a remote queue

A tutorial showing how to send messages to a remote queue manager.

Before you begin

This tutorial builds upon IBM MQ objects set up in [“Tutorial 1: Sending a message to a local queue”](#) on page 289. You must complete Tutorial 1 to proceed with this tutorial.

About this task



This tutorial shows you how to set up messaging between a queue manager that is called QM_ORANGE and a queue manager called QM_APPLE. You can complete this tutorial, and verify your environment, by setting up the sending queue manager on the same computer as the target queue manager. A message that is created on the sending queue manager is delivered to a queue called Q1 on the receiving queue manager (this queue is referred to as a *remote* queue).

Important: During this tutorial, you must use the computer on which you created queue manager QM_APPLE and local queue Q1.

You must set up a queue manager and queues (a remote queue definition and a transmission queue) on your computer, and then define a message channel. Finally, put a test message onto the sending queue manager, and get it from the queue on the receiving queue manager.

When you have completed this tutorial, you should have a basic understanding of how to set up and use IBM MQ messaging using a remote queue definition.

Creating the queue manager on the sending machine

Before you begin

Before creating a queue manager on the sending machine, you must ensure that IBM MQ Server is correctly installed. The queue manager on the sending machine cannot be created remotely. It must be created locally on the sending machine.

About this task

In this part of the tutorial, you will create a queue manager QM_ORANGE on the sending machine.

You can perform this task using either of the following interfaces:

- [The MQ Explorer graphical interface](#)
- [IBM MQ Script Command \(MQSC\) command-line interface](#)

Creating the queue manager using IBM MQ Explorer on the sending machine

About this task

This process cannot be done remotely. It must be done locally on the sending machine:

Procedure

1. Start IBM MQ Explorer.
2. In the **Navigator view**, right-click the **Queue Managers** folder, then click **New > Queue Manager**. The **Create Queue Manager** wizard opens.
3. In the **Queue Manager name** field, type QM_ORANGE.
4. Click **Next** twice to go to the Enter configuration options section of the wizard.
5. Select **Create server-connection channel**.
6. Ensure that **Automatic** is selected from the **Select type of queue manager startup** option.
7. Click **Next** to go to the **Enter listener options** section of the wizard.
8. Ensure that the **Create listener configured for TCP/IP** check box is selected.
9. If the **Finish** button is not available, type another port number in the **Listen on port number** field. If the current value is 1414, try typing 1415 or 1416.
10. Click **Finish**.

Results

An icon representing this queue manager is displayed in the **Queue Managers** folder in the **Navigator view** of IBM MQ Explorer, and the queue manager automatically starts running after you create it.

Creating the sending queue manager using MQSC

About this task

Open a command prompt on the sending machine and follow these steps:

Procedure

1. Create a default queue manager called QM_ORANGE by typing the command:

```
crtmqm QM_ORANGE
```

Messages tell you that the queue has been created and that the default IBM MQ objects have been created.

2. Start this queue manager by typing the command:

```
strmqm QM_ORANGE
```

A message tells you when the queue manager has started.

Results

You have now created the sending queue manager.

Creating the queues on the sending queue manager

Before you begin

Before creating the queues on the sending queue manager, you must already have created the queue manager in the task:

- [Creating the queue manager](#).

About this task

In this part of the tutorial, you will create a remote queue definition and a transmission queue on the sending queue manager.

You can perform this task using either of the following interfaces:

- [The IBM MQ Explorer graphical interface](#)
- [IBM MQ Script Command \(MQSC\) command-line interface](#)

Creating the queues on the sending queue manager using IBM MQ Explorer

About this task

On the sending queue manager:

Procedure

1. In the **Navigator view**, expand the **Queue Managers** folder.
2. Expand queue manager QM_ORANGE.
3. Right-click the **Queues** folder, then click **New > Remote Queue Definition**.
The **New Remote Queue Definition** wizard opens.
4. In the **Name** field, type Q1
5. Click **Next**.
6. In the **Remote queue** field, type Q1
7. In the **Remote queue manager** field, type QM_APPLE
8. In the **Transmission queue** field, type QM_APPLE
9. Click **Finish**.
You have now created the remote queue definition.
10. Click the QM_ORANGE queue manager.
11. Right-click the **Queues** folder, then click **New > Local Queue**.
The **New Local Queue** wizard opens.
12. In the **Name** field, type QM_APPLE
13. Click **Next**.
14. In the **Usage** field, select **Transmission**.
15. Click **Finish**.

You have now created the transmission queue on the local machine.

Results

The new queues, Q1 and QM_APPLE, are displayed in the **Content view**.

If the queues are not displayed in the **Content view**, click Refresh  in the **Content view**.

Creating the queues on the sending queue manager using MQSC

About this task

Open a command prompt on the sending machine and follow these steps:

Procedure

1. Start MQSC by typing the command:

```
runmqsc
```

A message tells you that an MQSC session has started.

2. Define a local queue called QM_APPLE by typing the following command:

```
define qlocal (QM_APPLE) usage (xmitq)
```

A message tells you when the queue has been created.

3. Define a remote queue definition by typing the following command:

```
define qremote (Q1) rname (Q1) rqmname(QM_APPLE) xmitq (QM_APPLE)
```

Results

You have now created the queues on the sending queue manager. The next task is to create the message channel between the sending and receiving queue managers.

Creating a message channel

Before you begin

Before creating a message channel, you must already have completed the following tasks in this tutorial:

- [Creating the queue manager](#)
- [Creating the queues](#)

About this task

In this part of the tutorial, you will create a message channel between the sending and receiving queue managers.

You can perform this task using either of the following interfaces:

- [The IBM MQ Explorer graphical interface](#)
- [IBM MQ Script Command \(MQSC\) command-line interface](#)

Creating the message channel using IBM MQ Explorer

Procedure

1. On the receiving queue manager QM_APPLE, create the receiver end of the channel:
 - a) In the **Navigator view**, expand the queue manager QM_APPLE that you created earlier.
 - b) Right-click the **Channels** folder, then click **New > Receiver Channel**.
The **New Receiver Channel** wizard opens.
 - c) In the **Name** field, type QM_ORANGE.QM_APPLE
 - d) Click **Finish**.

You have now created the receiver channel on the receiving machine.

2. On the sending queue manager QM_ORANGE, create the sender end of the channel:

- a) Expand the queue manager QM_ORANGE that you created earlier.
- b) Right-click the **Channels** folder, then click **New > Sender Channel**.

The **New Sender Channel** wizard opens.

- c) In the **Name** field, type QM_ORANGE.QM_APPLE, then click **Next**.
- d) In the **Connection name** field, type the computer name or IP address of the receiving machine (you should already have obtained this with your system administrator's help).

If the default port number 1414 was not used when creating QM_APPLE, the Connection name field entry should be of the format:

```
con-name(port)
```

Where con-name is the computer name or IP address of the receiving machine, and port is the port number used when the receiving queue manager was set up.

- e) In the **Transmission queue** field, type QM_APPLE

The transmission queue name you enter here must match the name you entered for the transmission queue in [Creating the queues on the sending queue manager](#).

- f) Click **Finish**.
- g) Click the **Channels** folder.
- h) Right-click **QM_ORANGE.QM_APPLE**.
- i) From the pop-up menu, click **Start**.
- j) Click **OK**.

You have now created the sender channel on the sending machine.

Note: You do not have to start the receiver channel because it started automatically when you set up the sender channel (when you set up the sender channel, you specified the receiver channel's IP address).

Results

You have now created a receiver channel QM_ORANGE.QM_APPLE, on the receiving queue manager QM_APPLE, and a sender channel QM_ORANGE.QM_APPLE, on the sending queue manager QM_ORANGE. You have also started the sender channel, which automatically started the receiver channel.

Creating the message channel using MQSC

Procedure

1. Open a command prompt on the receiving machine and follow these steps:

- a) Start MQSC by typing the command:

```
runmqsc
```

A message tells you that an MQSC session has started.

- b) Define a receiving channel by typing the following command:

```
define channel (QM_ORANGE.QM_APPLE) chltype (RCVR) trtype (TCP)
```

A message tells you when the channel has been created.

- c) Open a new command window and check which ports are free. Enter the following command:

```
netstat -an
```

This shows you a list of running processes. Check the port number of each of the processes to see if port 1414 is in use; you can find this by looking in the Local Address column. The information is given in the form ip_address:port_being_used.

If port 1414 is not in use, use 1414 as the port number for your listener and sender channel later in the verification. If it is in use, select an alternative port that is not in use; for example 1415 if this is not being used by another process.

- d) Verification requires you to start the default IBM MQ listener. By default, the listener will listen on port 1414. If you found that port 1414 was free in step c, no action is required and you can proceed to step e. If you must use a port other than 1414, alter the definition of the SYSTEM.DEFAULT.LISTENER.TCP. For example, to use port 1415, enter the following command in the MQSC window:

```
alter listener(system.default.listener.tcp) trptype(tcp) port(port_number)
```

Where port_number is the number of the port the listener should run on. This must be the same as the number used when defining your sender channel in step 2b of this procedure.

- e) In the MQSC window, start the default IBM MQ listener by entering the following command:

```
start listener(system.default.listener.tcp)
```

- f) Stop MQSC by typing:

```
end
```

Some messages are displayed followed by the command prompt.

2. Open a command prompt on the sending machine and follow these steps:

- a) Start MQSC by typing the command:

```
runmqsc
```

A message tells you that an MQSC session has started.

- b) Define a sender channel by typing the following command:

```
define channel(QM_ORANGE.QM_APPLE) chltype(sdr) conname('con-name(port)') xmitq(QM_APPLE)
trptype(tcp)
```

The value con-name is the TCP/IP address of the receiver workstation. The value port is the port on which the listener is running on the receiver machine, the default value is 1414.

- c) Start the channel by typing the following command:

```
start channel (QM_ORANGE.QM_APPLE)
```

- d) Stop MQSC by typing:

```
end
```

Some messages are displayed followed by the command prompt.

Results

You have now created all the IBM MQ objects required for messages to be sent from the sending queue manager QM_ORANGE to the queue Q1 on the receiving queue manager QM_APPLE. The next task is to send a test message.

Putting a test message on the queue

Put a test message on a remote queue using only the MQSC command-line interface.

Before you begin



Before putting a test message on the queue, you must already have completed the following tasks in this tutorial:

- [Creating the queue manager](#)
- [Creating the queues](#)
- [Creating a message channel](#)


About this task

Perform this task on the sending machine (the machine that hosts the queue manager QM_ORANGE).

The **amqsput** sample program is used to put a message onto the queue that you created:

-  **Windows** On Windows, the sample programs are installed by default with IBM MQ Server or Client.
-  **Linux** On Linux, the samples programs RPM need to be installed.

Procedure

1. Open a command prompt.
2. Start the **amqsput** sample program as follows:
 -  **Windows** On Windows, type the command:

```
amqsput Q1 QM_ORANGE
```

-  **Linux** On Linux, change to the /opt/mqm/samp/bin directory and type the command:

```
./amqsput Q1 QM_ORANGE
```

The following messages are displayed:

```
Sample amqsput0 start  
target queue is Q1
```

3. Type some message text on one or more lines, then press **Enter** twice.

The following message is displayed:

```
Sample amqsput0 end
```

Results

You have now created a test message and put it onto the remote queue. The next task is to verify that the test message was received.

Verifying that the test message was sent

Before you begin

Before getting the test message from the queue, you must already have completed the other tasks in this tutorial:

- [Creating the queue manager](#)
- [Creating the queues](#)

- [Creating a message channel](#)
- [Putting a test message on the queue](#)

About this task

This topic shows how to verify that the test message was sent.

You can perform this task using either of the following interfaces:

- [The IBM MQ Explorer graphical interface](#)
- [IBM MQ Script Command \(MQSC\) command-line interface](#)

Verifying that the test message was sent using IBM MQ Explorer

About this task

Perform this task on the receiving machine (the machine that hosts the queue manager QM_APPLE).

On the receiving queue manager:

Procedure

1. In the **Navigator view**, expand queue manager QM_APPLE.
2. Click the **Queues** folder.
3. In the **Content view**, right-click the queue Q1, then click **Browse Messages**.
The **Message browser** opens to show the list of the messages that are currently on Q1.
4. Double-click the last message in the list to view its properties dialog.

Results

On the **Data** page of the properties dialog, the **Message data** field displays the content of the message in human-readable form.

Verifying that the test message was sent using MQSC


About this task

You perform this task on the receiving machine (the machine that hosts the queue manager QM_APPLE). The amqsget sample program is used to get the message back from the queue.

Open a command prompt and follow these steps:

Procedure

Start the amqsget sample program as follows:

-  On Linux, change to the `MQ_INSTALLATION_PATH/samp/bin` directory, where `MQ_INSTALLATION_PATH` represents the high-level directory in which IBM MQ is installed. Type the command:

```
./amqsget Q1 QM_APPLE
```

-  On Windows, type the command:

```
amqsget Q1 QM_APPLE
```

Results

The sample program starts and your message is displayed along with any other messages on this queue. After a short pause, the sample program ends and the command prompt is displayed again.

You have now completed this tutorial.

Tutorial 3: Sending a message on a client-server configuration

A tutorial to set up messaging between client and server machines, sending a message from the client, and verifying its receipt.

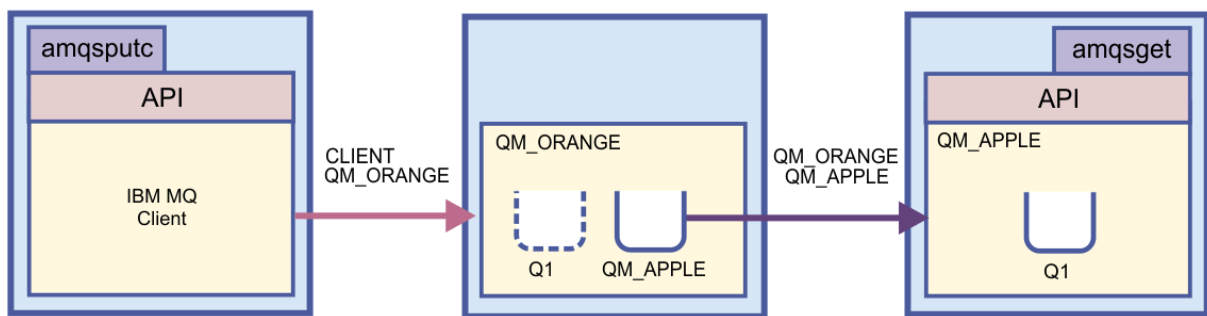
Before you begin

This tutorial builds upon IBM MQ objects that have been set up during the previous tutorial - you need to complete [“Tutorial 2: Sending a message to a remote queue”](#) on page 296 first.

Before starting this tutorial, you need to find out the name which identifies the server which hosts queue manager QM_ORANGE on the network from your system administrator.

You are currently in the IBM MQ Explorer help. To grant the client the access rights to put messages on to a queue, see *Preparing and running the sample programs* in the main product documentation.

About this task



This tutorial shows you how to set up messaging between client and server machines. From the client machine, you put a message on queue manager QM_ORANGE, which is hosted on a server machine. QM_ORANGE sends the message to Q1 on QM_APPLE, which is hosted on another server machine.

Important: This tutorial shows you how to work with a client-server installation, where the client is a third machine with IBM MQ Client installed, and the server is the machine which has the queue manager QM_ORANGE defined on it.

You set up the server by creating a server-connection channel. You then set up the client by defining the MQSERVER environment variable. Finally, you put a test message from the Client onto QM_ORANGE which sends it to queue Q1 on QM_APPLE and you verify that the message was sent.

When you have completed this tutorial, you should have a basic understanding of how to set up messaging on an IBM MQ MQI client-server configuration.

Setting up the server

About this task

In this part of the tutorial, you will set up the queue manager QM_ORANGE on the server machine to enable client connections to it. This will involve configuring a server-connection channel.

This task can be achieved by using either:

- [“Setting up the server using IBM MQ Explorer”](#) on page 305

- [“Setting up the server using MQSC” on page 305](#)

Setting up the server using IBM MQ Explorer

About this task

On the server machine that hosts queue manager QM_ORANGE:

Procedure

1. In the **Navigator view**, expand the **Queue Managers** folder.
2. Expand QM_ORANGE.
3. Right-click the **Channels** folder, then click **New > Server-connection Channel**.
The **New Server-connection Channel** wizard opens.
4. In the **Name** field, type CLIENT.QM_ORANGE then click **Next**.
5. Click **MCA** to open the **MCA** page.
6. In the **MCA User ID** field, type your Windows login name (or a user name in the mqm group).
7. Click **Finish**.

Results

The new server-connection channel is displayed in the **Content view**.

What to do next

For more information about the MCAUSER ID, see [Access control for clients](#) in the IBM MQ online product documentation.

Setting up the server using MQSC

About this task

Open a command prompt on the receiving machine and follow these steps:

Procedure


1. Start MQSC by typing the command:

```
runmqsc QM_ORANGE
```

A message tells you that an MQSC session has started. MQSC has no command prompt.

2. Define a server-connection channel by typing the following command on one line:

```
define channel(CLIENT.QM_ORANGE) chltype(SVRCONN) trptype(TCP) mcauser('mqm')
```

 **Windows** If you are using Windows, type your Windows login name (or a valid mqm user name) in place of mqm.

A message tells you when the channel has been created.

3. Stop MQSC by typing:

```
end
```

Some messages are displayed followed by the command prompt.

4. Start a listener by typing the following command:

```
runmqclsr -t tcp
```

Results

You have now finished setting up the server. The next task is to set up the client.

Setting up the client on Windows and Linux



Before you begin

Before setting up the client to communicate with queue manager QM_ORANGE, you must ensure that the IBM MQ MQI client has been installed on the client machine.

About this task

In this part of the tutorial, you must set up the client component on the server using the MQSERVER environment variable. You must find out the network name of the machine which hosts queue manager QM_ORANGE from your system administrator.

To set up the client on Windows or Linux, complete the steps for the platform that you are using:

-  [“Setting up the client on Windows” on page 306](#)
-  [“Setting up the client on Linux” on page 307](#)

Setting up the client on Windows

About this task

In this task, you will set up the client component using the MQSERVER environment variable. You will need to find out the network name of the machine which hosts queue manager QM_ORANGE from your system administrator.

Procedure

1. Open the Control Panel: Click **Start > Settings > Control Panel**
2. Double-click **System**.
3. Click the **Advanced** tab.
4. Click **Environment Variables**.
5. In the User Variables pane, click **New**.
6. Type MQSERVER into the Variable Name field.
7. Type CLIENT.QM_ORANGE/TCP/*hostname* into the Variable Value field, where *hostname* is the computer name or IP address that identifies the machine hosting queue manager QM_ORANGE. If you do not use the default port number 1414, you must also specify the port number where the listener is listening. For example: MQSERVER=CLIENT.QM_ORANGE/TCP/*hostname* (1415)
8. Click **OK**.

The MQSERVER environment variable is visible in the User Variables pane.

Results

You have now set up the client and server components needed on your Windows machine.

Setting up the client on Linux

About this task

In this task, you will set up the client component using the MQSERVER environment variable. You will need to find out the network name of the machine which hosts queue manager QM_ORANGE from your system administrator.

Procedure

1. Log in as the user who will be running Express File Transfer, who must be a member of the mqm group.
2. Open a command prompt
3. Type

```
cd $HOME
```

4. Use a text editor to edit the profile. This example assumes that you are using the bash shell, so you need to edit the file \$HOME/.bashrc. If you are using a different system shell, consult your system documentation. Add the following text to the end of the file:

```
MQSERVER=CLIENT.QM_ORANGE/TCP/'hostname'; export MQSERVER
```

Replace *hostname* with the name that identifies the server machine on the network.

5. Close the command prompt.
6. Log out and log back in for the change to take effect.

Results

You have now set up the client and server components needed. The next task is to send a message from the client to the server queue manager QM_ORANGE.

Sending a message from a client to a server


Before you begin

Before putting a test message on the queue, you must already have completed the following tasks in this tutorial:

- [“Setting up the server” on page 304](#)
- [“Setting up the client on Windows and Linux” on page 306.](#)

About this task

In this part of the tutorial, you will send a message from the client to the server queue manager QM_ORANGE, which uses the remote queue definition and other IBM MQ objects defined in earlier tutorials to route the message onto queue manager QM_APPLE and to queue Q1.


 On Windows, the sample programs are installed by default with IBM MQ Server or Client.

 On Linux, the sample programs RPM will need to be installed.

Open a command prompt on the client and follow these steps:

Procedure

1. Start the **amqspu** sample program as follows:

-  On Linux, change to the `MQ_INSTALLATION_PATH/samp/bin` directory, where `MQ_INSTALLATION_PATH` represents the high-level directory in which IBM MQ is installed. Type the command:

```
./amqsputc Q1
```

-  On Windows, type the command:

```
amqsputc Q1
```

The following messages are displayed:

```
Sample AMQSPUT0 start  
target queue is Q1
```

2. Type some message text on one or more lines, then press **Enter** twice.

The following message is displayed:

```
Sample AMQSPUT0 end
```

Results

You have now created a test message and sent it to the server queue manager QM_ORANGE, which routes it onto queue Q1 on queue manager QM_APPLE. The next task is to verify that the test message was received.

Verifying that the test message was sent

Before you begin

Before getting the test message from the queue, you must already have completed the following tasks in this tutorial:

- [“Setting up the server” on page 304](#)
- [“Setting up the client on Windows and Linux” on page 306](#)
- [“Sending a message from a client to a server” on page 307.](#)

About this task

This topic shows how to verify that the test message was sent.

You can perform this task using either of the following interfaces:

- [The IBM MQ Explorer graphical interface](#)
- [IBM MQ Script Command \(MQSC\) command-line interface](#)

Verifying that the test message was sent using IBM MQ Explorer

About this task

On the machine that hosts the queue manager QM_APPLE:

Procedure

1. In the **Navigator view**, expand QM_APPLE.
2. Click the **Queues** folder.
3. In the **Content view**, right-click Q1, then click **Browse Messages**.
The **Message browser** opens to show the list of messages on Q1.
4. Double-click the last message in the list to open its properties dialog.

Results

On the **Data** page of the properties dialog, the **Message data** field displays the content of the message in human-readable form.

Verifying that the test message was sent using MQSC


About this task

The **amqsget** sample program is used to get the message back from the queue.


Open a command prompt and start the **amqsget** sample program as follows:

Procedure

Start the **amqsget** sample program as follows:

-  On Linux, change to the `MQ_INSTALLATION_PATH/samp/bin` directory, where `MQ_INSTALLATION_PATH` represents the high-level directory in which IBM MQ is installed. Type the following command:

```
./amqsget Q1
```

-  On Windows, type the following command:

```
amqsget Q1
```

Results

The sample program starts, and your message is displayed along with any other messages on this queue. After a pause of 15 seconds, the sample ends and the command prompt is displayed again.

You have now completed this tutorial.

Reference

This section of the Help deals with reference material such as Accessibility, Properties, and Icons for IBM MQ Explorer.

The following topics list the reference material for IBM MQ Explorer.

- [Accessibility in IBM MQ Explorer](#)
- [Icons in IBM MQ Explorer](#)
- [Views in IBM MQ Explorer](#)
- [Properties](#)
- [Status attributes](#)
- [Byte array dialog](#)
- [Strings in property dialogs](#)

Accessibility in IBM MQ Explorer

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully.

When you use the administrative facilities that are provided by IBM MQ Explorer, you can use your operating system's accessibility features to modify the behavior of the user interface. You can change the key behavior, provide a high-contrast display, or control the pointer with keys instead of a mouse. For more information, see the documentation for your operating system.

The fully accessible way of using IBM MQ is to use the command-line interface. For more information, see [Accessibility features for IBM MQ](#) in the IBM MQ online product documentation.

IBM MQ Explorer has been designed with accessibility in mind. The following features have been implemented.

- IBM MQ Explorer uses icons to indicate the status of objects, such as queue managers. Screen readers cannot interpret these icons, so there is an option to show a textual description of the icon. To select this option, from within the IBM MQ Explorer click **Window > Preferences > MQ Explorer** and select **Show status of objects after object name**.
- In IBM MQ Explorer, there are several places where color is used to highlight parts of the interface. If you have difficulty interpreting the default colors, you can change them. For more information, see [“Changing the colors”](#) on page 254.
- In the **Manage Authority Records** dialogs, and in other dialogs that display object authorities, the tables use icons to show whether an authority is granted. Screen readers cannot interpret these icons. If you prefer, you can set the tables to use text instead of icons. For more information, see [“Displaying object authority settings as text”](#) on page 260.

Icons in IBM MQ Explorer

IBM MQ Explorer uses icons to represent the different objects, such as queue managers, queues, and channels.




IBM MQ Explorer alters the icons slightly to indicate the status of the objects, for example, running or stopped.


The tables on this page list the following types of icons:

- [Status](#)
- [Queue managers](#)
- [Queues](#)
- [Channels](#)
- [Other IBM MQ objects](#)
- [Queue manager clusters](#)
- [Queue sharing groups](#)
- [API exits](#)
- [JMS objects](#)

Status icons in IBM MQ Explorer

The following table lists the status icons that are superimposed on to IBM MQ object icons in IBM MQ Explorer to indicate the status of the object. To investigate the cause of an object being in an Alert or Warning state, look at the status of the current object.

Icon	Meaning
	Up. The object is running.
	Down. The object is not running.
	Alert. The status of the object is indeterminate; for example, the object is in the process of stopping or starting.







Icon	Meaning
	Warning. The object is having problems connecting. On queue managers in the Full Repository and Partial Repository folders, this icon means that the queue manager is suspended from the cluster.

Queue managers

The following table lists the icons that are used in IBM MQ Explorer to represent queue managers.




The queue manager icon is yellow when IBM MQ Explorer is connected to a queue manager; when it is not connected, the icon is gray. Local queue managers are marked with an Up or Down icon to show whether the queue manager is running or stopped.




Remote queue managers are distinct because their icons do not indicate whether they are running or not; they indicate only whether they are connected to IBM MQ Explorer. A remote queue manager must be running for IBM MQ Explorer to connect to it; if IBM MQ Explorer is disconnected from a remote queue manager, it is not possible for IBM MQ Explorer to detect whether the remote queue manager is running.

Icon	Queue manager local or remote to Explorer?	Explorer connected?	Queue manager status
	Local	Yes	Running
	Local	No	Running
	Local	No	Running as standby
	Local	No	Stopped
	Remote	Yes	Running
	Remote	No	Unknown

Queue manager instances









Display queue manager instance status by selecting a remote queue manager in the navigator and clicking **Connection details > Manage instances**.

Icon	Connection status text	Explanation
	Connected	The IBM MQ Explorer is connected and the queue manager is running.
	Not connected	The IBM MQ Explorer has not attempted to connect to the queue manager instance. Click Test connections to update the status of the connection.
	Not connection	The instance is running as a standby.

Icon	Connection status text	Explanation
	Not available	One of three reasons: <ul style="list-style-type: none"> • Unknown host name or IP address. • The queue manager is not listening on the port address. • The IBM MQ Explorer has timed out waiting for a response from the queue manager instance.
	Different name	The queue manager instance listening on the IP address configured for the connection has a different queue manager name.
	Different UUID	The queue manager instance listening on this IP address has a different UUID. The problem might be because the listener is connected to a different queue manager with the same name, rather than another instance of the same queue manager. It might also be because the remote queue manager has been deleted and re-created with the same name. It is no longer the same queue manager.









Queues

The following table lists the icons that are use in IBM MQ Explorer to represent queues.

Icon	Meaning
	Local
	Local queue that is shared in a cluster
	Model
	Remote queue definition
	Remote queue definition of a queue that is shared in a cluster
	Alias
	Alias queue that is shared in a cluster
	Transmission

Channels










The following table lists the icons that are used in IBM MQ Explorer to represent channels.

Icon	Meaning
	Sender
	Server
	Receiver
	Requester
	Server-connection
	Client-connection
	Cluster-sender
	Cluster-receiver

Other IBM MQ objects






The following table lists the icons that are used in IBM MQ Explorer to represent other IBM MQ objects.









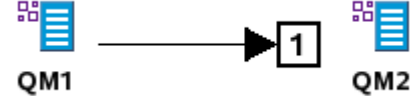

Custom services can be in either running, stopped, alert, or warning states.

Icon	Meaning
	Topic
	Subscription
	Listener
	Authentication information object
	Namelist
	Process definition
	Message
	Custom service
	Application connection

Queue manager clusters











The following table lists the icons that are used in IBM MQ Explorer to represent clusters.





Icon	Meaning
	Cluster
	Cluster without a source provider
	Full repository
	Full repository suspended from the cluster
	Full repository disconnected

Icon	Meaning
	Partial repository
	Partial repository suspended from the cluster
	Partial repository disconnected
	Cluster-receiver channel
	Cluster-sender channel
	Local queue that is shared in a cluster
	Remote queue definition of a queue that is shared in a cluster
	The number indicates the number of cluster queues in the cluster.
	The number indicates the number of instances of cluster-sender channels between the two queue managers.
	The number indicates the number of instances of cluster-receiver channels on the queue manager.

Queue-sharing groups




The following table lists the icons that are used in IBM MQ Explorer to represent queue sharing groups. The channels can be in either running, stopped, alert, or warning states.

Icon	Meaning
	Queue sharing group
	QSG authentication information object
	QSG local queue
	QSG model queue
	QSG namelist
	QSG process definition
	QSG alias queue
	QSG receiver channel
	QSG remote queue definition
	QSG requester channel

Icon	Meaning
	QSG sender channel
	QSG server channel
	QSG server-connection channel
	QSG transmission queue










API Exits

The following table lists the icons that are used in IBM MQ Explorer to represent API exits.

Icon	Meaning
	Common
	Template
	Local

JMS objects

The following table lists the icons that are used in IBM MQ Explorer to represent JMS objects in the JNDI namespace.

Header	Header
	Initial context; connected
	Initial context; disconnected
	Connection factory for MQ connections
	Connection factory for realtime connections
	Destination object for a queue
	Destination object for a topic
	Subcontext; connected
	Subcontext; disconnected
	Corrupted JNDI object

Views in IBM MQ Explorer

IBM MQ Explorer is a perspective in Eclipse Platform, which is built on Eclipse technology.

The IBM MQ Explorer perspective is a collection of views. The IBM MQ Explorer perspective contains two main views:

1. [“IBM MQ Explorer Navigator view” on page 316](#)
2. [“IBM MQ Explorer Content view” on page 324](#)

Depending on which other plug-ins you have installed and enabled, the IBM MQ Explorer perspective might also contain other views. You can also show any other available views in the IBM MQ Explorer perspective by clicking **Window > Show Views > Other...** and selecting a view.

Related tasks

[“Configuring IBM MQ Explorer” on page 223](#)

Use this information to help you to configure your IBM MQ Explorer installation.

IBM MQ Explorer Navigator view



The Navigator view in IBM MQ Explorer displays all of the IBM MQ objects that you can administer and monitor in IBM MQ Explorer.


The types of objects that you can administer and monitor in the Navigator view include objects that are on other computers and on other platforms, such as AIX, Linux, and z/OS.



Objects and folders in the Navigator view









The Navigator view contains a hierarchy of objects and folders that contain the queue managers and their objects. The following table contains descriptions of the objects and folders that are shown in the Navigator view.





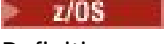
Object or folder	Purpose of the object or folder	Tasks that you can perform	Links to more information
IBM MQ	The IBM MQ object is the root of the folder hierarchy and represents the installation of IBM MQ on the computer.	Right-click the IBM MQ object to perform tasks that affect the whole of IBM MQ on the local computer, such as configuring IBM MQ properties, starting trace, or managing TLS certificates.	Configuring IBM MQ
Queue Managers	The Queue Managers folder contains all of the queue managers that are on the local computer unless you have hidden them. You can also add remote queue managers, including z/OS queue managers. You can administer and monitor any queue manager using IBM MQ as long as it is shown in the Queue Managers folder.	Click the Queue Managers folder to list the queue managers, and their attributes, in the Content view. Right-click the Queue Managers folder to perform tasks such as creating a new queue manager, or adding a remote queue manager to IBM MQ Explorer.	Queue managers

Object or folder	Purpose of the object or folder	Tasks that you can perform	Links to more information
A queue manager	Each queue manager that is shown in the Queue Managers folder is represented by a queue manager object icon in the hierarchy. The objects that belong to a queue manager are organized in folders under that queue manager.	Click the queue manager to display an overview of the queue manager's attributes in the Content view. Right-click the queue manager to perform tasks such as starting and stopping the queue manager, or configuring the queue manager's properties. Expand the queue manager to display the folders that contain the queue manager's objects.	Queue managers
Queues	The Queues folder contains all of the queues that are defined on the queue manager.	Click the Queues folder to list the queue manager's queues in the Content view. Right-click the Queues folder to perform tasks such as creating a new queue.	Queues
Channels	The Channels folder contains all of the channels that are defined on the queue manager except for the client-connection channels.	Click the Channels folder to list the queue manager's channels in the Content view. Right-click the Channels folder to perform tasks such as creating new channels.	Channels
Client Connections	The Client Connections folder contains all of the client-connection channels that are defined on the queue manager.	Click the Client Connections folder to list the queue manager's client-connection channels in the Content view. Right-click the Client Connections folder to perform tasks such as creating a new client-connection channel.	Channels
  Listeners	The Listeners folder contains all of the listener objects that you have defined on the queue manager. If you start a listener service from the command line, that listener is not shown in the Listeners folder. The Listeners folder is not available on z/OS queue managers.	Click the Listeners folder to list the queue manager's listener objects in the Content view. Right-click the Listeners folder to perform tasks such as creating a new listener object.	Listeners

Object or folder	Purpose of the object or folder	Tasks that you can perform	Links to more information
 Multi Services	The Services folder contains all of the custom services that you have defined on the queue manager. The Services folder is not available on z/OS queue managers.	Click the Services folder to list the queue manager's custom services in the Content view. Right-click the Services folder to perform tasks such as creating a new custom service.	Services
Process Definitions	The Process Definitions folder contains all of the process definitions that are defined on the queue manager. A process definition contains the details of an application that starts in response to a trigger event on a queue manager.	Click the Process Definitions folder to list the process definitions in the Content view. Right-click the Process Definitions folder to perform tasks such as creating a new process definition.	Process definitions
Namelist	The Namelist folder contains all of the namelists that are defined on the queue manager. A namelist is a list of the names of other MQ objects.	Click the Namelist folder to list the namelists in the Content view. Right-click the Namelist folder to perform tasks such as creating a new namelist.	Namelists

Object or folder	Purpose of the object or folder	Tasks that you can perform	Links to more information
Authentication Information	<p>The Authentication Information folder contains all of the authentication information objects that are defined on the queue manager. There are two types of authentication objects:</p> <ul style="list-style-type: none"> • CRL LDAP authentication object contains the authentication information that is used to connect to LDAP servers that hold Certificate Revocation Lists (CRLs). The queue manager connects to the CRL LDAP servers when transporting data that is encrypted using TLS. •   On UNIX and Windows systems, IBM MQ TLS support can check for revoked certificates using OCSP (Online Certificate Status Protocol). OCSP is the preferred method. IBM MQ classes for Java and IBM MQ classes for Java cannot use the OCSP information in a client channel definition table file. However, you can configure OCSP as described in Clustering: Using REFRESH CLUSTER best practices in the online product documentation. 	Click the Authentication Information folder to list the authentication information objects in the Content view. Right-click the Authentication Information folder to perform tasks such as creating a new authentication object.	Authentication information

Object or folder	Purpose of the object or folder	Tasks that you can perform	Links to more information
  Storage Class	The Storage Class folder contains all of the storage classes that are on the queue manager. The messages that are on queues that are not shared are stored on page sets. Storage classes map the queues to page sets. The Storage Class folder is available only on z/OS queue managers.	Click the Storage Class folder to list the storage classes in the Content view. Right-click the Storage Class folder to perform tasks such as creating a new storage class.	Storage class
  Queue sharing groups	The Queue sharing groups folder contains all of the queue sharing groups that are defined on the queue manager. The Queue sharing groups folder is available only on z/OS queue managers.	Click the Queue-sharing Groups folder to list the queue sharing groups in the Content view. Expand the Queue-sharing Groups folder to display the queue sharing groups that it contains.	Queue sharing groups
  A queue sharing group	Each queue sharing group that is shown in the Queue sharing groups folder is represented by a node in the hierarchy. Objects in a queue sharing group are available to all of the queue managers that belong to the queue-sharing group. The objects that are in the queue sharing group are organized in folders under that queue sharing group. Queue sharing groups are available only on z/OS queue managers.	Click the queue sharing group to display an overview of the queue sharing group's attributes in the Content view. Right-click the queue sharing group to perform tasks such as deleting the queue-sharing group, or viewing the queue sharing group's properties. Expand the queue sharing group to display the folders that contain the objects that are available to all of the queue managers in the queue-sharing group.	Queue sharing groups
  Shared Queues	The Shared Queues folder contains all of the queues that have the disposition Shared in the queue sharing group. All of the queue managers in the queue-sharing group can put and get from the shared queues without needing channels. The Shared Queues folder is available only on z/OS queue managers.	Click the Shared Queues folder to list the shared queues in the Content view. Right-click the Shared Queues folder to perform tasks such as creating a new shared queue.	“IBM MQ queues” on page 29

Object or folder	Purpose of the object or folder	Tasks that you can perform	Links to more information
  Coupling Facility Structures	<p>The Coupling Facility Structures folder contains all of the coupling facility structure objects that are available in the queue sharing group. Coupling facility objects represent physical coupling facility structures, which store the messages that are on shared queues so that any queue manager in the queue sharing group can get the messages. The Coupling Facility Structures folder is available only on z/OS queue managers.</p>	<p>Click the Coupling Facility Structures folder to list the coupling facility objects in the Content view. Right-click the Coupling Facility Structures folder to perform tasks such as creating a new coupling facility object to represent a physical coupling facility structure.</p>	 Coupling facility structures
  Group Definitions	<p>The Group Definitions folder does not directly contain any MQ objects. The Group Definitions folder contains other folders that contain MQ objects in the queue sharing group that have the disposition Group, which means that all the queue managers in the queue sharing group have a copy of the objects. The objects that are in folders in the Group Definitions folder are a subset of the objects that belong to the z/OS queue manager. You cannot hide the Group Definitions folder. The Group Definitions folder is available only on z/OS queue managers.</p>	<p>Expand the Group Definitions folder to display the object folders that it contains.</p>	Queue sharing groups

Object or folder	Purpose of the object or folder	Tasks that you can perform	Links to more information
Queue Manager Clusters	The Queue Manager Clusters folder contains all of the clusters that IBM MQ Explorer knows about. IBM MQ Explorer knows about clusters that belong to queue managers in the Queue Managers folder. The Queue Manager Clusters folder is available only when a queue manager in the <u>Queue Managers</u> folder belongs to a cluster.	Click the Queue Manager Clusters folder to list the clusters in the Content view. Right-click the Queue Manager Clusters folder to perform tasks such as creating a new queue manager cluster.	Queue manager clusters
A cluster	Each queue manager cluster that is shown in the Queue Manager Clusters folder is represented by a node in the hierarchy. The queue managers that belong to the cluster are shown in folders in the Queue Manager Clusters folder. Queue manager clusters are available to administer in IBM MQ Explorer only when a queue manager in the Queue Managers folder belongs to a cluster.	Click the queue manager cluster to view information about it. Right-click the queue manager cluster to perform tasks such as adding queue managers to it.	Queue manager clusters
Full Repositories	The Full Repositories folder contains all of the queue managers that host full repositories of the cluster. The queue managers in the Full Repositories folder store a complete, up-to-date set of information about the cluster. The Full Repositories folder is available only when a queue manager in the Queue Managers folder belongs to a cluster.	Click the Full Repositories folder to list the queue managers that host full repositories for the cluster in the Content view. Right-click the Full Repositories folder to add another queue manager that hosts a full repository to the cluster.	Queue manager clusters

Object or folder	Purpose of the object or folder	Tasks that you can perform	Links to more information
A queue manager that hosts a full repository	Each queue manager that hosts a full repository for the cluster is represented by a node in the hierarchy. The queue managers in the Full Repositories folder store a complete, up-to-date set of information about the cluster.	Click the queue manager to list the cluster queues and cluster channels that are available on the queue managers. Right-click the queue manager to perform tasks such as removing the queue manager from the cluster, or share the queue manager's queues in the cluster.	Queue manager clusters
Partial Repositories	The Partial Repositories folder contains all of the queue managers that host partial repositories of the cluster. The queue managers in the Partial Repositories folder store only information about objects in the cluster that those queue managers need to used.	Click the Partial Repositories folder to list the queue managers that host partial repositories for the cluster in the Content view. Right-click the Partial Repositories folder to add another queue manager that hosts a partial repository to the cluster.	Queue manager clusters
A queue manager that hosts a partial repository	Each queue manager that hosts a partial repository for the cluster is represented by a node in the hierarchy. The queue managers in the Partial Repositories folder store only information about objects in the cluster that those queue managers need to used.	Click the queue manager to list the cluster queues and cluster channels that are available on the queue managers. Right-click the queue manager to perform tasks such as removing the queue manager from the cluster, or share the queue manager's queues in the cluster.	Queue manager clusters

Depending on which other plug-ins you have installed and enabled for IBM MQ Explorer, the Navigator view might contain other folders and objects.

Related tasks

[“Showing or hiding a queue manager” on page 98](#)

By default, the Navigator view shows all of the queue managers on the computer on which IBM MQ Explorer is installed. However, if you have any queue managers that you are not currently administering, you can, if you wish, choose to hide them. You can also show and hide remote queue managers.

[“Enabling installed plug-ins” on page 255](#)

If a new plug-in that you install in IBM MQ Explorer is not enabled by default, you can enable it by using the Preferences dialog.

Related reference

[“Icons in IBM MQ Explorer” on page 310](#)

IBM MQ Explorer uses icons to represent the different objects, such as queue managers, queues, and channels.

[“Views in IBM MQ Explorer” on page 315](#)

IBM MQ Explorer is a perspective in Eclipse Platform, which is built on Eclipse technology.

IBM MQ Explorer Content view

The Content view in IBM MQ Explorer displays information about objects and properties.

When you click a folder in the **Navigator** view, the **Content** view displays the IBM MQ objects, and their properties, that are in the folder. When you click a queue manager in the **Navigator** view, a summary, or QuickView, of the queue manager's properties is displayed in the **Content** view.

When you select a queue manager set in the **Navigator** view, the **Content** view displays the grouped queue managers, as well as if the set is automatic or manual.

You can customize and reorder the properties and objects that are displayed in the **Content** view by using [schemes](#) and [filters](#).

Gray cells in the Content view

Empty gray cells in the **Content** view indicate that the property is not relevant and cannot be set. For example, the Scope attribute is not valid for model queues. If you click the **Queues** folder to display its contents in the **Content** view, the Scope attribute cell is shaded gray for any model queues. Similarly, the Transmission Queue attribute cell is shaded gray for all queue types except for remote queue definitions, for which you can set the Transmission Queue attribute. If you have not set the Transmission Queue attribute for a remote queue definition, the cell is empty and white. White cells indicate that the property can be set.

You can change the color of these cells in the **Preferences** dialog. For more information, see [“Changing the colors”](#) on page 254.

Related concepts

[“Define schemes to change the order of columns in tables”](#) on page 250

When object data is displayed in IBM MQ Explorer in tables, you can customized the order of the columns in the tables.

Related tasks

[“Filtering the objects displayed in tables”](#) on page 225

When object data is displayed in IBM MQ Explorer in tables, you can filter the data so that only the objects in which you are interested are displayed.

[“Configuring IBM MQ Explorer”](#) on page 223

Use this information to help you to configure your IBM MQ Explorer installation.

[“Enabling installed plug-ins”](#) on page 255

If a new plug-in that you install in IBM MQ Explorer is not enabled by default, you can enable it by using the Preferences dialog.

Related reference

[“Icons in IBM MQ Explorer”](#) on page 310

IBM MQ Explorer uses icons to represent the different objects, such as queue managers, queues, and channels.

[“Views in IBM MQ Explorer”](#) on page 315

IBM MQ Explorer is a perspective in Eclipse Platform, which is built on Eclipse technology.

Preferences for IBM MQ Explorer

Use this information to help you to configure your IBM MQ Explorer installation.

You can customize the way that IBM MQ Explorer presents information to you, including the order of columns in tables and the objects that are displayed in the **Content** view. You can also customize a range of other preferences to suit your business needs. The availability of certain preferences depends on which plugins you have that are installed and enabled for IBM MQ Explorer.

Key settings can be configured in the **Preferences** dialog. To open the **Preferences** dialog: Click **Window** > **Preferences**. Other settings can be configured as per the information in the following table:

<i>Table 5. Configuring IBM MQ Explorer preferences</i>		
To configure	Task	See
Authorization service	Displaying object authority settings as text	“Displaying object authority settings as text” on page 260
Client connections	Remote queue managers; specifying default values that are used to connect to remote queue managers	“Specifying the default values used to connect to remote queue managers” on page 256
	TLS Key Repositories; specifying the default location and default password of TLS certificates	“Specifying the default location and default password of TLS certificates” on page 105
	TLS Options; specifying default security preferences	“Default security preferences” on page 192
	Security exit; configuring a default security exit	“Configuring a default security exit” on page 190
	User identification; enabling default user identification	“Users and groups (entities) in the authorization service” on page 179
Display settings	Changing the colors	“Changing the colors” on page 254
	Defining schemes and filters from the relevant content view	Filtering the objects displayed in the Content view
	Setting the order of columns in tables and the objects that are displayed	Changing the order of columns in tables
	Changing the refresh frequency of queue manager information	“Changing the refresh frequency of queue manager information” on page 255
	Displaying object authority settings as text	“Displaying object authority settings as text” on page 260
Enable Plug-ins	Enabling installed plug-ins	“Enabling installed plug-ins” on page 255
Managed File Transfer	Configuring managed file transfer	“Configuring Managed File Transfer preferences” on page 330
Messages	Configuring messages	“Configuring message preferences” on page 332
Passwords	Setting password preferences	“Passwords preferences” on page 194
Telemetry	Configuring telemetry channels	“Telemetry channels” on page 270

Table 5. Configuring IBM MQ Explorer preferences (continued)

To configure	Task	See
Tests	Including hidden queue managers in test configurations	“Including hidden queue managers in test configurations” on page 259
	Including SYSTEM objects when you run tests	“Including SYSTEM objects when you run tests” on page 259

Filtering the objects displayed in tables

When object data is displayed in IBM MQ Explorer in tables, you can filter the data so that only the objects in which you are interested are displayed.

About this task

An example of the filtering that you can do is that when the Content view displays the contents of a queue manager's **Queues** folder, you might choose to display only the queues that have a name that starts with **saturn**. Instead of creating a new filter for each type of object on each queue manager, you can create a filter for a type of object, such as queues, on one queue manager, and then reuse the filter for queues on other queue managers.

You can filter on the following criteria:

- The name of the object.
- The type of the object (queues and channels only).
- An attribute of the object (optional).

IBM MQ Explorer supplies and applies a standard filter for each type of object. For example, the **Standard for Queues** filter includes all the queues so that you can be sure that you are viewing all the queues on the queue manager, the **Standard for Channels** filter includes all the channels, and so on. IBM MQ also supplies a selection of other useful filters; for example, an **All queues with messages** filter displays only queues that have one or more messages.

To apply a different filter to a folder of objects:

Procedure

1. In the Content view or dialog that contains the table, click the small arrow next to the current filter name. A menu is displayed.
2. If you want to apply one of the other supplied filters, in the menu, click the name of the filter. The menu closes and the filter is applied to the table.
3. If you want to apply a different filter (that was not supplied with IBM MQ), click **More Filters...** The Select Filter dialog opens displaying the filters that are available.
4. In the **Apply filter** list, click the filter that you want to apply, or click **No filter** to remove all filtering from the table.
5. Click **OK**.

Results

The selected filter is applied to the selected folder.

Related concepts

[“Define schemes to change the order of columns in tables” on page 250](#)


When object data is displayed in IBM MQ Explorer in tables, you can customized the order of the columns in the tables.

Define schemes to change the order of columns in tables

When object data is displayed in IBM MQ Explorer in tables, you can customized the order of the columns in the tables.

For example, queues have a large number of attributes; when you display the contents of a **Queues** folder in the **Content** view, if you are not interested the creation date and time of the queues, you can choose not to display those attributes, or you can move those columns to somewhere else in the table.

The changes that you make are saved as schemes so that you can apply the same set of changes to the same type of objects on other queue managers too.

 IBM MQ Explorer supplies and applies standard schemes. Because IBM MQ for z/OS for queue managers and objects can have slightly different attributes, each object scheme has settings for the object on Multiplatform queue managers and for z/OS queue managers. The standard schemes include all the attributes for objects of that type. For example, the **Standard for Queues** scheme includes all the attributes for queues on Multiplatforms and z/OS platforms so that you can be sure that you can see all the attributes for the queues that are listed.

To apply an existing scheme to a table:

1. In the **Content** view, or dialog that contains the table, click the small arrow next to the current scheme name. A menu is displayed.
2. From the menu, click **Select Scheme**. The **Select Scheme** dialog opens.
3. In the **Select Scheme** dialog, click the scheme that you want to apply. The attributes that the scheme will display are listed in the dialog.
4. Click **OK**.

The selected scheme is applied to the folder of objects.

Related tasks

[“Creating a scheme” on page 251](#)

You can create schemes for most of the tables of data in IBM MQ Explorer.

[“Editing an existing scheme” on page 252](#)

You can edit any schemes that you have created previously and you can also edit the schemes that are supplied with IBM MQ Explorer, for example, the **Standard for Queues** scheme.

[“Copying an existing scheme” on page 253](#)

If there already exists a scheme that is similar to a scheme that you want to create, you can copy the existing scheme and then edit it as required.

[“Filtering the objects displayed in tables” on page 225](#)

When object data is displayed in IBM MQ Explorer in tables, you can filter the data so that only the objects in which you are interested are displayed.

Changing the refresh frequency of queue manager information

In IBM MQ Explorer, the information about local and remote queue managers is automatically refreshed at set intervals. You can change the refresh frequency for a specific queue manager, or the default refresh frequency for all new queue managers. You can also prevent a specific queue manager's information from being refreshed automatically.

About this task

By default, the information that is displayed about remote queue managers is refreshed less frequently than the information about local queue managers because of the increase in network traffic every time that the information is requested from remote systems.

You can also prevent specific queue managers being refreshed automatically. For example, if you know that a queue manager will remain stopped for a period of time, you could prevent IBM MQ Explorer requesting information about it and so reduce the network traffic.

- [“Changing the refresh frequency for a specific queue manager” on page 256](#)
- [“Changing the default refresh frequency for all new queue managers” on page 256](#)
- [“Preventing automatic refresh of a queue manager's information” on page 256](#)

Changing the refresh frequency for a specific queue manager

About this task

To change the frequency with which information about a specific queue manager is refreshed:

Procedure

1. In the **Navigator** view, right-click the queue manager, then click **Connection Details > Set Refresh Interval**. The **Automatic Refresh** dialog opens.
2. In the **Automatic Refresh** dialog, edit the value in the **Interval** field.
3. Optional: To reset the automatic refresh rate to the default value, click **Apply Default**.
4. Click **OK** to save the new refresh rate.

Results

The information about the queue manager is now automatically refreshed at the new rate.

Changing the default refresh frequency for all new queue managers

About this task

To change the frequency with which the information about local and remote queue managers is refreshed:

Procedure

1. Click **Window > Preferences** to open the Preferences dialog.
2. On the **MQ Explorer** page, in the **Default Queue Manager Refresh Intervals** fields, type the refresh interval, in seconds, then click **OK**.

Results

All new queue managers that are added to IBM MQ Explorer are now refreshed at the new rate.

Preventing automatic refresh of a queue manager's information

About this task

To prevent queue manager information in IBM MQ Explorer from being refreshed automatically:

Procedure

1. In the **Navigator** view, right-click the queue manager, then click **Connection Details > Set Refresh Interval**. The **Automatic Refresh** dialog opens.
2. In the **Automatic Refresh** dialog, clear the check box, then click **OK**.

Results

The information about the queue manager is no longer refreshed automatically. To refresh the information about the queue manager, click **Refresh** on the menu in the **Content** view.

Displaying object authority settings as text

In IBM MQ Explorer dialogs, object authorities are displayed as icons by default, but you can choose to have them displayed as text instead.

About this task

In the Manage Authority Records dialogs, and in other dialogs that display object authorities, the tables use icons to show whether an authority is granted. If you prefer, you can set the tables to use text instead of icons.

To change the tables to use text instead of icons to show whether an authority is granted, complete the following steps.

Procedure

1. Open the Preferences dialog: **Window > Preferences**
2. Expand **MQ Explorer**.
3. On the **Authorization Service** page, click **Display authorities as text**.
4. Click **OK** to close the Preferences dialog.

Results

The next time that you open a dialog that displays object authorities, the tables will show authorities using text instead of icons.

Related tasks

[“Configuring IBM MQ Explorer” on page 223](#)

Use this information to help you to configure your IBM MQ Explorer installation.

Related reference

[“Accessibility in IBM MQ Explorer” on page 309](#)

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully.

Changing the colors

You can change the colors that are used to highlight parts of the IBM MQ Explorer interface.

About this task

In IBM MQ Explorer, there are several places where color is used to highlight parts of the interface. For example, in the Content view, cells that are not applicable to an object are colored gray; in the command window that contains the command details of the Create Queue Manager wizard, sections of the text are highlighted in different colors. In the Preferences dialog, you can change the colors that are used.

To change the color of cells that are not applicable:

Procedure

1. Open the Preferences dialog: **Window > Preferences**
2. In navigation tree of the Preferences dialog, expand **MQ Explorer**, then click **Colors**.
3. On the **Colors** page, click the palette button for the feature that you want to change. The palette button in the Content View section of the page controls the color of cells that are not applicable (cells that are colored gray by default); the palette buttons in the Command Details section of the page control the color of the text and background in the command windows that are displayed in the Details window when you create, delete, start, and stop a queue manager in IBM MQ Explorer.
4. In the palette, click the color that you want to use (or define a custom color), then click **OK**.

5. Click **OK** to close the Preferences dialog.

Results

The color that you selected is used.

Related tasks

[“Configuring IBM MQ Explorer” on page 223](#)

Use this information to help you to configure your IBM MQ Explorer installation.

Related reference

[“Accessibility in IBM MQ Explorer” on page 309](#)

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully.

Enabling installed plug-ins

If a new plug-in that you install in IBM MQ Explorer is not enabled by default, you can enable it by using the Preferences dialog.

About this task

When you install a new plug-in in IBM MQ Explorer, whether the plug-in is supplied by IBM or by a third party, if the plugin does not appear to be working in IBM MQ Explorer, it is likely that the plug-in is not enabled by default.

To enable an installed plugin, complete the following steps.

Procedure

1. Click **Window > Preferences** to open the Preferences dialog.
2. In the navigation tree of the **Preferences** dialog, expand **MQ Explorer**, then click **Enable plug-ins**. A list of the available plugins is displayed.
3. Select the check box next to the plug-in that you want to enable, then click **OK**.

Results

The plug-in is now enabled in IBM MQ Explorer. Any folders or menu items for example, that are related to the plug-in are now available in IBM MQ Explorer.

You can also disable plug-ins that you do not use. For example, if you do not use clustering in your messaging networks, you can clear the check box next to the Cluster Component plugin. The Cluster Component plugin remains installed on your computer so that you can enable it in future. Because the plug-in is still installed on your computer, the help that is associated with clustering is still available in the help system and in the context-sensitive help.

Configuring Managed File Transfer preferences

The Managed File Transfer preferences that you can configure in IBM MQ Explorer include general preferences and the default global configuration type. You can also select the level of functional fix pack behavior that you want to enable.

About this task

To configure Managed File Transfer preferences within IBM MQ Explorer, complete one of the following tasks:

- [“Setting general preferences” on page 331](#)
- [“Overriding the default global configuration subscription type” on page 331](#)

- [“Selecting the level of fix pack function” on page 331](#)

Setting general preferences

About this task

You can override the default time zone in which to display any date and time values in the Managed File Transfer panels, and you can select the maximum number of log and progress messages to store and retain in the plugin's state.

Procedure

1. Click **Window**, and then click **Preferences**.
The **Preferences** dialog opens.
2. Click **Managed File Transfer**.
Managed file transfer settings appear.
3. To override the default time zone in which to display any date and time values in the Managed File Transfer panels, choose your preferred time zone from the **Time zone** list.
4. To choose the maximum number of log and progress messages to store, select the appropriate values. Defaults for each are 1000.

Overriding the default global configuration subscription type

About this task

You can override the type of subscription that is made when you are connecting to a globally configured property set. You can configure subscriptions to be durable or non-durable. This setting determines what happens when the plug-in disconnects from the coordination queue manager. Durable subscriptions continue to exist while the plug-in is disconnected and continue to receive publications. Non-durable subscriptions exist only if the connection to the queue manager is available.

Procedure

1. Click **Window**, and then click **Preferences**.
The **Preferences** dialog opens.
2. Click **Managed File Transfer**.
Managed file transfer settings appear.
3. Under **Default global configuration subscription type**, choose from either **Durable** or **Non-durable**.

Selecting the level of fix pack function

About this task

You can select the level of functional fix pack behavior that you want to enable. In addition to APAR fixes, certain levels of IBM MQ Explorer Fix Packs contain new function or changes in behavior that are disabled by default. To enable these changes, select the level of function that you want to move to. This setting does not affect the application of fixes, which are always applied.

Procedure

1. Click **Window**, and then click **Preferences**.
The **Preferences** dialog opens.
2. Click **Managed File Transfer**.
Managed file transfer settings are displayed.

3. Select the level of function to which you want to move.

Configuring message preferences

You can configure browsing limits and set how message properties are displayed.

About this task

To configure message preferences within IBM MQ Explorer, complete one of the following tasks:

- [“Configuring browsing limits” on page 332](#)
- [“Showing message properties” on page 332](#)

Configuring browsing limits

About this task

You can configure the following values:

- The maximum number of messages that can be browsed in the range 1 - 5000.
- The maximum number of bytes of data to be displayed per message in the range 0 - 16 384.

Important: Ensure that your system has enough random access memory (RAM) to display large numbers of messages or system performance might be affected.

Procedure

1. Click **Window**, and then click **Preferences**.
The **Preferences** dialog opens.
2. Click **Messages**.
Messages settings appear.
3. Change the maximum number of messages that are browsed by either clicking the up or down arrows, or by typing a new value. The default value is 500.
4. Change the maximum data bytes that are displayed by either clicking the up or down arrows, or by typing a new value. The default value is 1000.

Showing message properties

About this task

You can set how message properties are displayed. You can show no message properties, or you can show message properties as Named Properties, as an MQRFH2 structure in the message body, or as an MQRFH2 structure in the message body compatible with IBM WebSphere MQ 6.0.

Procedure

1. Click **Window**, and then click **Preferences**.
The **Preferences** dialog opens.
2. Click **Messages**.
Messages settings appear.
3. To show no message properties, except those properties that are contained in the message descriptor or extension, clear the **Show message properties** check box.
For more information, see [“Named Properties page” on page 555](#).
4. To show message properties as Named Properties, select the **as Named Properties** check box.
Properties of the message, except those properties that are contained in the message descriptor or extension, are represented in the **Named Properties** panel in name-value pairs, and the properties are removed from the message data.

For more information, see the entry for MQGMO_PROPERTIES_IN_HANDLE in “Named Properties page” on page 555.

5. To show message properties as an MQRFH2 structure in the message body, select the **as an MQRFH2 structure in message body** check box. Properties of the message, except those properties that are contained in the message descriptor or extension, are represented in the **MQRFH2 Properties** panel and the properties remain in the message data.

For more information, see the entry for MQGMO_PROPERTIES_FORCE_MQRFH2 in “MQRFH2 Properties page” on page 555.

6. To show message properties as an MQRFH2 structure in the message body, select the **as an MQRFH2 structure in message body, compatible with WebSphere MQ V6** check box. If the message contains a property with a prefix of mcd . , jms . , usr . , or mqext . , all message properties, except those properties that are contained in the message descriptor or extension, are represented in the **MQRFH2 Properties** panel and the properties remain in the message data. Otherwise, all properties of the message, except those properties that are contained in the message descriptor or extension, are discarded and not displayed.

For more information, see the entry for MQGMO_PROPERTIES_FORCE_MQRFH2 in “MQRFH2 Properties page” on page 555.

User identification

The user identification for all queue managers in a set can be changed. The user identification can be overridden when you add a new remote queue manager.

The user identification preferences are part of the **Preferences** dialog, and they can be opened in the following way:

1. Click **Windows > Preferences....** The **Preferences** dialog opens.
2. Expand **MQ Explorer**.
3. Expand **User identification**. The default user identification settings dialogs are now accessible.

Select **Enable default user identification** to enable the **Userid** and **Password** fields.

Item	Description
Enable user identification	Select Enable user identification to enable the fields on this dialog.
User identification compatibility Mode	When selected, the userid and password are passed to the server in a way compatible with security exits created prior to IBM MQ 8.0.
Userid	The userid and password, when specified, are passed to the server, and can be used either by: <ul style="list-style-type: none">• The queue manager, if configured to use connection authentication, or• A server security exit, if using a client connection to establish the identity of the IBM MQ Explorer user.
No password	When selected, no password is passed to the server with the userid.
Prompt for password	When selected, the user is prompted for a password that is passed to the server with the userid. The prompting occurs as part of the connect operation.

Item	Description
Use saved password	When selected, the saved password is passed to the server with the userid.
Saved password	The saved password to be passed to the server with the userid

Related reference

[“Default security preferences” on page 192](#)

A security exit can be defined for all client connections in the same IBM MQ Explorer. This is known as a default security exit and the preferences for the security exit are described here.

[“Passwords preferences” on page 194](#)

You can store passwords to a file so that you do not have to enter them every time you want to connect to resources.

Passwords preferences

You can store passwords to a file so that you do not have to enter them every time you want to connect to resources.

Passwords used by the IBM MQ Explorer to connect to resources (for example: opening TLS stores or connecting to queue managers), can be stored in a file. The password file can be stored locally, to a remote device, or to a removable device.

To open the **Passwords** preference panel:

1. Click **Window > Preferences**. The **Preferences** dialog opens.
2. Expand **MQ Explorer**.
3. Select **Passwords** to display the **Passwords** panel.

Item	Description
Do not save passwords	Passwords are not stored to a file. This is the default value.
Save passwords to file	Passwords are saved to the file you specify. Select Save passwords to file and click Browse to select a location for the encrypted password file
Use default key	You must use a key to open a password store. This is the default value.
User defined key	You must use a key to open a password store. Select User defined key then click Change to enter your password. The password must contain a minimum of 8 characters.

Related tasks

[“Configuring a default security exit” on page 190](#)

A security exit can be defined for all client connections in the same IBM MQ Explorer. This is known as a default security exit.

[“Configuring the client security details for a queue manager set” on page 191](#)

The client security details and security exit can be defined for all the client-connected queue managers in a queue manager set.

Related reference

[“Default security preferences” on page 192](#)

A security exit can be defined for all client connections in the same IBM MQ Explorer. This is known as a default security exit and the preferences for the security exit are described here.

Default security preferences

A security exit can be defined for all client connections in the same IBM MQ Explorer. This is known as a default security exit and the preferences for the security exit are described here.

The default security preferences are part of the **Preferences** dialog, and they can be opened in the following way:

1. Click **Windows > Preferences....** The **Preferences** dialog opens.
2. Expand **MQ Explorer**.
3. Expand **Client Connections**. The default security settings dialogs are now accessible.

Security Exit

Select **Enable default security exit** to set the default security exit for all client connections in the same IBM MQ Explorer. The security exit for all the client-connected queue managers in a set can be changed. The security exit can be overridden if you define a new security exit when you add a new remote queue manager.

The Security Exit for all client-connected queue managers in a set can be changed. The TLS options can be overridden when you add a new remote queue manager.

Item	Description
Exit name	Specifies the name of the exit program to be run by the security exit. Exit name can be up to 1024 characters long and is case sensitive. Exit name can be a fully qualified java class name found in the directory or jar file. Exit name can be a C exit, of the format: <code>dll_name(function_name)</code> . The default path for exits is always used to locate C exits, you cannot specify the location of the exit library in this entry field unless no default path is set.
in directory	Specifies the directory for the security exit (Java exits only).
in jar	Specifies the jar file for the security exit (Java exits only).
Exit data	Exit data can be up to 32 characters long. If no value has been defined for that attribute, this field is all blanks.

SSL/TLS Options

Select **Enable default SSL options** to enable the default SSL/TLS options for all client connections in the same IBM MQ Explorer. The SSL/TLS options for all client-connected queue managers in a set can be changed. The SSL/TLS options can be overridden when you add a new remote queue manager.

Item	Description
SSL CipherSpec	<p>The CipherSpec identifies the combination of encryption algorithm and hash function used by an SSL/TLS connection. A CipherSpec forms part of a CipherSuite, which identifies the key exchange and authentication mechanism as well as the encryption and hash function algorithms.</p> <p>The size of the key used during the handshake can depend on the digital certificate you use, but some of the CipherSpecs supported by IBM MQ include a specification of the handshake key size. Note that larger handshake key sizes provide stronger authentication. With smaller key sizes, the handshake is faster.</p> <p>For more information, see CipherSpecs and CipherSuites in the IBM MQ online product documentation.</p>

Item	Description
SSL FIPS required	<p>Select Yes to use only FIPS-certified cipher suites. If you select Yes, then all TLS connections must use FIPS-certified cipher suites.</p> <p>Select No to use any available cipher suites.</p> <p>The default setting is No.</p> <p>If you change this setting from Yes to No, or from No to Yes a dialog will be opened asking if you want to restart MQ Explorer.</p> <p>Any changes to this setting will not be applied until the MQ Explorer has been restarted.</p>
SSL reset count	<p>Type the number of bytes, from 0 to 999 999 999, that are sent and received within a TLS conversation before the secret key is renegotiated. A value of 0 means that the secret key is never renegotiated. The number of bytes includes control information that is sent by the message channel agent (MCA). If the value of this attribute is greater than 0 and the value of the Heartbeat interval attribute in the Channel properties is greater than 0, the secret key is also renegotiated before message data is sent or received following a channel heartbeat.</p>
Peer name	<p>The Distinguished Name (DN) of the queue manager to be used by TLS. The peer name is set to indicate that connections will only be allowed where the server is successfully authenticated as a specific DN.</p>

SSL/TLS Stores

Select **Enable default SSL stores** to work with the Trusted Certificate Store and the Personal Certificate Store.

To configure IBM MQ Explorer with the location and password of the SSL/TLS certificate store, refer to: [“Specifying the default location and default password of TLS certificates” on page 105.](#)

By enabling the default SSL/TLS stores, IBM MQ Explorer can use the certificates in the TrustStore and KeyStore to connect to remote queue managers with a TLS-enabled connection.

The SSL/TLS Stores for all client-connected queue managers in a set can be changed. The SSL/TLS Stores can be overridden when you add a new remote queue manager.

Related tasks

[“Configuring a default security exit” on page 190](#)

A security exit can be defined for all client connections in the same IBM MQ Explorer. This is known as a default security exit.

[“Configuring the client security details for a queue manager set” on page 191](#)

The client security details and security exit can be defined for all the client-connected queue managers in a queue manager set.

Related reference

[“Passwords preferences” on page 194](#)

You can store passwords to a file so that you do not have to enter them every time you want to connect to resources.

Configuring a default security exit

A security exit can be defined for all client connections in the same IBM MQ Explorer. This is known as a default security exit.

About this task

The default security definitions will be persisted in IBM MQ Explorer and will be automatically included in **Preferences** in any import actions or export actions. The security exit details for each queue manager will be persisted with the other connection details of the queue manager.

To configure the default security exit:

Procedure

1. Click **Window > Preferences**.
The **Preferences** dialog opens.
2. Expand **MQ Explorer**.
3. Expand **Client Connections**.
The default security settings dialogs are now accessible.
4. Configure the security settings as required.

What to do next

The default security exit has now been configured. All new client connections in the same IBM MQ Explorer now use the settings you have configured as a default. The settings can be overridden when adding a new remote queue manager.

Related tasks

[“Configuring the client security details for a queue manager set” on page 191](#)

The client security details and security exit can be defined for all the client-connected queue managers in a queue manager set.

Related reference

[“Default security preferences” on page 192](#)

A security exit can be defined for all client connections in the same IBM MQ Explorer. This is known as a default security exit and the preferences for the security exit are described here.

[“Passwords preferences” on page 194](#)

You can store passwords to a file so that you do not have to enter them every time you want to connect to resources.

Specifying the default location and default password of TLS certificates

You can configure IBM MQ Explorer to use TLS certificates in the TrustStore and KeyStore to connect to remote queue managers with an TLS-enabled connection.

About this task

To configure IBM MQ Explorer with the location and password of the TLS certificate store, complete the following tasks in IBM MQ Explorer on the computer from which you want to connect to the remote queue manager:

Procedure

1. In IBM MQ Explorer, click **Window > Preferences**.
The **Preferences** dialog opens.
2. Expand **MQ Explorer**.
3. Expand **Client Connections**. The default security settings dialogs are now accessible.
4. Select **SSL Key Repositories** to display the **SSL Key Repositories** pane.
5. In the **Trusted Certificate Store** field, browse for the location of the TrustStore on the computer, and in **Personal Certificate Store** field, browse for the location of the KeyStore on the computer.
The TrustStore and KeyStore contain the TLS certificates that are used with connections using client channel definition tables. It is possible that the TrustStore and KeyStore are in the same location on your computer.
6. (Optional) Click **Enter password...** in the Trusted certificate Store section to open the **SSL Password** dialog; in the **SSL Password** dialog, type the password that IBM MQ Explorer will need to access the store.

7. Click **Enter password...** in the Personal Certificate Store section to open the **SSL Password** dialog; in the **SSL Password** dialog, type the password that IBM MQ Explorer will need to access the store.
8. Click **OK** to save your changes and to close the Preferences dialog.

Results

IBM MQ Explorer can now use the TLS certificates in the TrustStore and KeyStore to connect to remote queue managers with an TLS-enabled connection.

Related tasks

[“Showing a remote queue manager” on page 99](#)

If you want to administer a remote queue manager, you must connect IBM MQ Explorer to the remote queue manager, so that the queue manager then appears in the Navigator view.

[“Creating a client channel definition table” on page 104](#)

You can create a client channel definition table for a queue manager to make it easier to connect instances of IBM MQ Explorer to the queue manager.

Related reference

[“Default security preferences” on page 192](#)

A security exit can be defined for all client connections in the same IBM MQ Explorer. This is known as a default security exit and the preferences for the security exit are described here.

Telemetry channels

A Telemetry channel is a communication link between a queue manager on IBM MQ, and MQTT clients. Each channel might have one or more telemetry devices connected to it.

For messages flowing from IBM MQ to MQTT clients, messages are taken from the default MQTT transmit queue, and sent through the telemetry channel. Messages destined for specific MQTT clients are routed to them using their client identifiers.

Advanced option

Telemetry channels have an option which sets the maximum number of client connections that can be displayed in the **Channel Status Content** view. This option is called **Max responses**. The default value is 500. Consider configuring this option before you start your queue manager. If your queue manager is running, you must restart it to apply the advanced option changes.

To configure the maximum responses option, perform the following actions:

1. Click **Window > Preferences**.
2. Expand **IBM MQ Explorer**, then click **Telemetry**.
3. In the **Max responses** field, type the number of client connections to display at any one time.
4. Click **OK**.

Client connections on all telemetry channels up to the maximum response limit are shown in the **Channel Status Content** view. If client connections exceed this limit, a warning is displayed within the **Content** view. For example, if you set the maximum responses to 10 and you reach or exceed this number, the following warning is displayed: The display has been limited to the first 10 responses. Use a filter to select a subset of responses.

The **Telemetry channel status window** shows client connections specific to that channel. The maximum response option limit applies only to client connections on this channel.

Related tasks

[“Creating and configuring a telemetry channel” on page 275](#)

A telemetry channel connects a number of MQTT clients to IBM MQ. Create one or more telemetry channels on a queue manager. Each of these telemetry channels might have different configuration settings, making it easier to manage the clients attached to them.

[“Starting and stopping a telemetry channel” on page 280](#)

[“Viewing the status of a telemetry channel” on page 281](#)

[“Filtering Telemetry objects” on page 282](#)

If you are viewing several defined telemetry objects in the **Content** view, you might need a way to narrow the search scope of these objects. Do this by using filters.

Including hidden queue managers in test configurations

Queue managers that are hidden in IBM MQ Explorer do not by default appear in the list of available objects when you create new test configurations. However, you can choose to have hidden queue managers included so that they listed as available queue managers against which you can run tests.

About this task

Queue managers that are currently hidden in IBM MQ Explorer are not currently of interest to you so, by default, they are not included in the list of available objects when you create new test configurations.

To include hidden queue managers, complete the following steps.

Procedure

1. Click **Window > Preferences** to open the Preferences dialog.
2. In the navigation tree of the Preferences dialog, expand **IBM MQ Explorer**, then click **Tests**.
3. Select the **Include hidden objects in the list of available objects** check box.

Results

Next time you create or edit a test configuration, any hidden queue managers are listed as available queue managers against which you can run the tests.

Including SYSTEM objects when you run tests

By default, SYSTEM objects are not included in test results, but you can choose to include them if required.

About this task

Definitions of SYSTEM.DEFAULT objects are provided in IBM MQ as incomplete templates so, by default, they are not included when you run tests. However, you can include them if you want.

To include SYSTEM objects in test results, complete the following steps.

Procedure

1. Click **Window > Preferences** to open the Preferences dialog.
2. In the navigation tree of the Preferences dialog, expand **IBM MQ Explorer**, then click **Tests**.
3. Select the **Include SYSTEM objects in the test results** check box.

Results

Next time you run tests against objects in IBM MQ Explorer, any available SYSTEM objects are also tested.




Properties

Use this information to find out about the properties that you can view and edit, including properties that apply to the whole IBM MQ installation and the properties of an individual IBM MQ object such as a queue, a queue manager, or a channel.

In IBM MQ Explorer, right-click any IBM MQ object, for example, a queue, a queue manager, or a channel, then click **Properties** to view and edit the object's properties. The properties are displayed in a properties

dialog that is divided into pages according to the type of the properties, for example, TLS, exits, and clusters.

The following topics list all of the properties for IBM MQ objects. For each property, there is a description of how to use it and why you might want to set it. The topics also include, where relevant, the equivalent MQI call that you can use when programming applications, and the equivalent MQSC command that you can enter on the command line.

- [IBM MQ](#)
- [Queue managers](#)
- [Queues](#)
- [Channels, including client connections](#)
- [Listeners](#)
- [Topics](#)
- [Services](#)
- [Service definitions](#)
- [Subscriptions](#)
- [Process definitions](#)
- [Namelists](#)
- [Authentication information](#)
- [“Channel authentication record properties” on page 501](#)
-  [Storage classes](#)
-  [Queue sharing groups](#)
-  [Coupling Facility structures](#)
- [Cluster queue manager](#)
- [Cluster queue](#)
- [Cluster topic](#)
- [Application connection](#)
- [Messages](#)
- [JMS connection factories](#)
- [JMS destinations](#)

Related tasks

[“Configuring queue managers and objects” on page 51](#)

You can configure many of the attributes of queue managers and their objects from IBM MQ Explorer using the properties dialogs.

IBM MQ properties

IBM MQ properties apply to the whole IBM MQ installation.

The following tables list the properties that you can set for IBM MQ:

- [General](#)
- [Extended](#)
- [Exits](#)
- [Default log settings](#)
- [ACPI](#)
- [Alert monitor](#)

- [Configuration information](#)

For each property, there is a brief description of when you might need to configure the property. The attributes in the IBM MQ properties dialog relate to stanzas in the configuration files.

General page

The following table lists the properties that you can set on the **General** page of the Properties for IBM MQ dialog.

Property	Description	Stanza key
Default prefix	To change the location of the directory that stores all the queue manager data, type the full path to the new directory.	DefaultPrefix
Default queue manager name	To specify a default name for new queue managers, type the name in this field.	Name

Extended page

The following table lists the properties that you can set on the **Extended** page of the Properties for IBM MQ dialog.

Property	Description	Stanza key
Convert EBCDIC newline	EBCDIC code pages contain a newline (NL) character that is not supported by the ASCII code pages (although some ISO variants of ASCII contain an equivalent). If messages are sent from a system that uses EBCDIC code pages (for example, a z/OS system) to a system that uses ASCII, you can control how the EBCDIC newline character is converted into ASCII format. The default value is NL_TO_LF , which means that the EBCDIC NL character (X'15') is converted to the ASCII line feed character LF (X'0A') for all EBCDIC to ASCII conversions. To convert the EBCDIC NL character according to the conversion tables on your operating system, click TABLE . Note that the results of a TABLE conversion can vary from platform to platform and from language to language; even on the same platform the results might vary if you use different coded character set identifiers (CCSIDs). To convert ISO CCSIDs using the TABLE method and use the NL_TO_LF method for all other CCSIDs, click ISO .	ConvEBCDICNewline

Property	Description	Stanza key
Cluster workload mode	The cluster workload exit, CLWL, allows you to specify which cluster queue in the cluster is to be opened in response to an MQI call (for example MQOPEN or MQPUT). The default value is SAFE , which means that the CLWL exit is run in a separate process to the queue manager so that if there is a problem, the integrity of the queue manager is preserved. However running the CLWL exit as a separate process can have a detrimental effect on performance. To improve performance by running the CLWL exit in the same process as the queue manager, click FAST . Use FAST mode only if you are certain that there are no problems with your CLWL exit because if there is a problem in FAST mode, the queue manager fails and the queue manager's integrity is at risk. This value can be overridden for individual queue managers using the cluster workload mode attribute. For more information, see “Queue manager properties” on page 349 .	CLWLMode

Exits page

The following table lists the properties that you can set on the **Exits** page of the Properties for IBM MQ dialog. To configure the exits that are common to all of the queue managers on this computer, edit the attributes on the **Exits** page.

Property	Description	Stanza key
Exits Default Path	To change the location of channel exits for clients, and channel exits and data conversion exits for servers, type the path to the new directory.	ExitsDefaultPath
Common API Exits	To configure a new common API exit for IBM MQ, click Add then in the Properties dialog, type the details of the exit. To edit a common API exit that is already shown in the table, click Edit ; to remove an API exit from the table, click Remove .	ApiExitCommon

Property	Description	Stanza key
Template API Exits	To configure a new template API exit for IBM MQ, click Add then in the Properties dialog, type the details of the exit. To edit a template API exit that is already shown in the table, click Edit ; to remove an API exit from the table, click Remove .	ApiExitTemplate
Name	Specifies the descriptive name of the API exit which is passed to the API exit in the ExitInfoName field of the MQAXP structure. This name must be unique and is limited to a length of 48 characters, and must contain only those characters that are valid for the name of IBM MQ objects, such as queue names.	Name
Type	Specifies the type of exit: common, or template.	(Not a separate stanza key.)
Sequence	This attribute is an unsigned numeric value that defines the sequence in which this API exit is called in comparison with other API exits. An API exit with a low sequence number is called before another API exit with a higher sequence number. The order in which different API exits with the same sequence number are called is undefined. It is perfectly value for there to be gaps in the sequence numbers of the API exits defined for a queue manager.	Sequence
Module	Specifies the module that contains the code for the API exit. If this field contains the full path name of the module it is used as is. If this field contains just the module name, the module is located using the same method as channel exits; that is, using the value in the Exit default path field on the Exits page of the queue manager properties dialog.	Module
Function	Specifies the name of the function entry-point into the module that contains the code for the API exit. This entry-point is the MQ_INIT_EXIT function. The length of this field is limited to MQ_EXIT_NAME_LENGTH.	Function

Property	Description	Stanza key
Data	If this attribute is specified, leading and trailing blanks are removed, the remaining string is truncated to 32 characters and the result is passed to the exit in the ExitData field of the MQAXP structure. If this attribute is not specified, the default value of 32 blanks is passed to the exit in the ExitData field of the MQAXP structure.	Data



Default log settings

The following table lists the properties that you can set on the **Default Log Settings** page of the Properties for IBM MQ dialog. To change the default log settings, edit the attributes on the **Default Log Settings** page. These settings are applied to all new queue managers by default.

Property	Description	Stanza key
Log type	To enable the queue manager to recover when it is restarted, click Circular . If the queue manager uses circular logging, when the log file is full, the file is overwritten starting from the beginning of the file. To enable the queue manager to recover when it is restarted and to enable media or forward recovery, click Linear . If the queue manager uses linear logging, when the log file is full, a new log file is started.	LogType
Log path	To change the default location of the logs, type the full path here. If you do not specify the path here, the default is a sub-directory called Log in the DefaultPrefix, where DefaultPrefix is the directory specified in the Default prefix property on the General page of the Properties for IBM MQ dialog.	LogPath
Log file pages	Type the number, from 32 - 4095, of 4KB pages in the log file. For example, if you type 256, the file size is 1MB.	LogFilePages

Property	Description	Stanza key
Log primary files	<p>On Windows, type the number, from 2 to 254, of primary log files. The default value is 3. The total number of primary and secondary log files must not exceed 255 and must not be less than 3.</p> <p>On UNIX and Linux, type the number, from 2 to 510, of primary log files. The default value is 3. The total number of primary and secondary log files must not exceed 511 and must not be less than 3.</p>	LogPrimaryFiles
Log secondary files	<p>On Windows, type the number, from 1 to 253, of secondary log files. The default value is 3. The total number of primary and secondary log files must not exceed 255 and must not be less than 3.</p> <p>On UNIX and Linux, type the number, from 1 to 509, of secondary log files. The default value is 3. The total number of primary and secondary log files must not exceed 511 and must not be less than 3.</p>	LogSecondaryFiles
Log buffer pages	<p>Type the number, from 0 - 512, of 4KB buffer pages for writing. If you specify 0, the queue manager selects the number itself.</p> <p>If you type a number from 1 to 17, the minimum of 18 is used. If you type a number from 18 - 512, that number of pages is used. If you change the value of this property, restart the queue manager to detect the change.</p>	LogBufferPages

Property	Description	Stanza key
Log write integrity	<p>The method that the logger uses to reliably write log records.</p> <p>The default value is TripleWrite. Note, that you can select DoubleWrite, but if you do so, the system interprets this as TripleWrite.</p> <p>You should use SingleWrite, only if the file-system or device hosting the IBM MQ recovery log explicitly guarantees the atomicity of 4KB writes.</p> <p>That is, when a write of a 4KB page fails for any reason, the only two possible states are either the before image, or the after image. No intermediate state should be possible.</p>	LogWriteIntegrity

Property	Description	Stanza key
  Log management	<p>The method used to manage your logs. LogManagement applies only when LogType is LINEAR.</p> <p>If you change the LogManagement value, the change does not take effect until the queue manager is restarted.</p> <p>There are three options.</p> <p>Manual, where you manage the log extents manually. Specifying this option means that the queue manager does not reuse or delete log extents, even when they are no longer required for recovery.</p> <p>Automatic, where log extents are managed automatically by the queue manager. Specifying this option means that the queue manager is able to reuse or delete log extents as soon as they are no longer required for recovery. No allowance is made for archiving.</p> <p>Archive, where log extents are managed by the queue manager, but you must notify the queue manager when archiving of each log extent is complete.</p> <p>Specifying this option means that the queue manager is free to reuse or delete a log extent, as soon as the queue manager has been notified that an extent no longer required for recovery has been archived.</p> <p>The default value is Manual.</p>	LogManagement

ACPI page

The following table lists the properties that you can set on the **ACPI** page of the Properties for IBM MQ dialog. ACPI (Advanced Configuration and Power® Interface) is an operating system feature that allows the computer to detect certain activity states and consequently to hibernate, that is to switch itself into a low power mode with no programs running, and in such a manner as to allow a quick "wake up".

When ACPI wants to put the computer into hibernation it first sends a suspend request to all applications. To control how IBM MQ responds to this request, set the **Do dialog** property on the **ACPI** page.

Property	Description	Stanza key
Do dialog	The default value is Yes , which means that IBM MQ displays a message that asks the user whether to suspend the running queue managers. To suspend IBM MQ without displaying this message, click No .	DoDialog
Deny suspend	If the Do dialog property is not set, or if it is set but the dialog cannot be displayed (for example, if a laptop computer has its lid closed) then Deny suspend controls the response. The default value is No , which means that IBM MQ suspends, even if the dialog cannot be displayed. To prevent IBM MQ suspending when the dialog cannot be displayed, click Yes . This property can be overruled by the Check channels running property.	DenySuspend
Check channels running	The default value is No , which means that IBM MQ does not check whether there are any channels running, and responds as directed by the Do dialog and Deny suspend properties. To check whether there are any channels running, click Yes . If there are no channels running, IBM MQ ignores the Do dialog and Deny suspend properties. If there are channels running, IBM MQ responds as directed by the Do dialog and Deny suspend properties.	CheckChannelsRunning

Alert monitor page

Windows

The Alert monitor is available only on Windows.

The following table lists the properties that you can set on the **Alert monitor** page of the Properties for IBM MQ dialog. The Alert monitor is useful for problem determination. Alerts are raised by the services when something goes wrong; for example, if a channel initiator service cannot start because a queue that is needed has been deleted. To configure the alert monitor, edit the attributes on the **Alert monitor** page.

Property	Description	Stanza key
Alert monitor notifies user	The default value is No , which means that IBM MQ does not send alerts to the user when there is a problem. To configure IBM MQ to send alerts when there is a problem, click Yes .	Enable

Property	Description	Stanza key
Alert monitor user	Type the computer name or the user name to which IBM MQ should send alerts.	Recipient
Alert monitor icon added to taskbar	The default value is No , which means that the Alert Monitor icon is not shown in the Windows system tray. To show the Alert Monitor icon in the Windows system tray, click Yes .	TaskBar

Configuration information page

The following table lists the properties that are displayed on the **Configuration information** page of the Properties for IBM MQ dialog. The properties on the **Configuration information** page are read-only.

Property	Description
Install type	Read-only. This attribute indicates whether you have installed the Server or Client version of IBM MQ on this computer.
mqjbnd05 loaded	Read-only. This is the library that is required to connect to local queue managers.
MQ Version	Read-only. This is the version of IBM MQ installed on this computer.
Build level	Read-only. This is the build number of the IBM MQ product that is installed on this computer.
Build type	Read-only. This is the type of build of the IBM MQ product that is installed on this computer.

Related tasks

[“Configuring IBM MQ using IBM MQ Explorer” on page 26](#)

In the Navigator view, you can use the Properties dialog to configure certain IBM MQ properties that apply to the whole installation. If necessary, you can also configure the properties of individual queue managers.

Queue manager properties











You can set attributes for both local and remote queue managers.


The tables on this page list all the attributes that you can set for local and remote queue managers. For each attribute, there is a brief description of when you might need to configure the attribute. Where relevant, the tables also give the equivalent MQSC parameter for the **ALTER QMGR** and **DISPLAY QMGR** commands. For more information about MQSC commands, see [Script \(MQSC\) Commands](#) in the online IBM MQ product documentation.


Those attributes for a queue manager that are changed through MQSC using the **ALTER QMGR** commands are the ones that are shown for both local and remote queue managers. Note, that the IBM MQ Explorer does not show all the properties for remote queue managers.

Those attributes that are defined in the `qm.ini` file, are the ones that are shown only for local queue managers. For example, the specification for the Recovery Logs and the XA apply to the `qm.ini` file, and so, is shown only for the local queue manager.


For lists of all the attributes that you can set for both local and remote queue managers in the queue manager's properties dialog see the following tables:

- [General](#)
- [Extended](#)
-  [Exits \(Multiplatforms\)](#)
- [Cluster](#)
- [Repository](#)
- [Communication](#)
- [Events](#)
- [SSL](#)
- [Statistics](#)
- [Online Monitoring](#)
-  [Statistics Monitoring \(Multiplatforms\)](#)
-  [Accounting Monitoring \(Multiplatforms\)](#)
-  [Log \(Multiplatforms\)](#)
-  [XA resource managers \(Multiplatforms\)](#)
-  [Installable services \(Multiplatforms\)](#)
- [Channels](#)
-  [TCP \(Multiplatforms\)](#)
-  [LU6.2 \(Multiplatforms\)](#)
-  [NetBIOS \(Multiplatforms\)](#)
-  [SPX \(Multiplatforms\)](#)
- [Publish/Subscribe](#)





 Some of these properties pages are available only on Multiplatforms queue managers, such as those running on Windows, AIX, or Linux (x86 and x86-64 platforms).

 Properties pages that are not available on z/OS queue managers are indicated.

The attributes that are marked with an asterisk (*) update configuration files so you can view and edit them when the queue manager is stopped. If you edit the marked attributes when the queue manager is running, you must stop and restart the queue manager so that the changes take effect. You can edit the unmarked attributes only when the queue manager is running. For more information about the configuration attributes, see [Changing queue manager configuration information](#) in the IBM MQ online product documentation.

 The following tables list the system parameters that you can set for remote z/OS queue managers. These attributes are not displayed in the Queue Manager properties dialog. They are included here because they are still attributes of the queue manager. For more information, see [Configuring z/OS queue manager system parameters](#).

-  [Archive \(z/OS\)](#)
-  [Archive tape \(z/OS\)](#)
-  [Statistics monitoring \(z/OS\)](#)
-  [Accounting monitoring \(z/OS\)](#)
-  [Log \(z/OS\)](#)

-  [Log copy \(z/OS\)](#)
-  [Security \(z/OS\)](#)
-  [Securityswitch \(z/OS\)](#)
-  [System \(z/OS\)](#)

For more information, see [Administering IBM MQ](#) and [Script \(MQSC\) Commands](#) in the IBM MQ online product documentation.

General

The following table lists the attributes that you can set on the **General** page of the Queue Manager properties dialog. The attributes marked with an asterisk (*) on the **General** page relate to stanzas in the configuration files.

Attribute	Meaning	MQSC parameter
*Queue manager name	Read-only. You cannot change the name of the queue manager after it has been created.	QMNAME
*Platform	Read-only. This is the architecture of the platform on which the queue manager is running.	PLATFORM
Queue manager status	Read-only. This attribute shows the status of the queue manager, which can be 1 of the following options: 1. Running 2. Starting 3. Quiescing	STATUS
Coded character set ID	Read-only. This is the coded character set identifier (CCSID) for the queue manager. The CCSID is the identifier that is used with all character string fields defined by the API. The value must be defined for use on your platform and must use a character set that is appropriate to the platform.	CCSID
Description	Type a meaningful description of the purpose of the queue manager. See Entering strings in IBM MQ Explorer .	DESCR
*Command level	Read-only. This is the function level of the queue manager.	CMDLEVEL






Attribute	Meaning	MQSC parameter
Version	<p>Read only. This is the version of IBM MQ installed. The format is VVRRMMFF:</p> <ul style="list-style-type: none"> • VV: Version • RR: Release • MM: Maintenance level • FF: Fix level 	VERSION
<div>Windows</div> <div>Windows</div> *Startup	<p>The Startup attribute controls how the selected queue manager is started. This attribute applies to Windows only. There are four options for the Startup attribute.</p> <p>Select Automatic to start the queue manager automatically when the IBM MQ Series service starts. This is the default value.</p> <p>Select Automatic, permitting multiple instances of the queue manager, to start the queue manager automatically when the IBM MQ Series service starts. For more information, see the <i>sax</i> option of <i>crtmqm</i> in the IBM MQ online product documentation.</p> <p>Select Interactive (manual) to start the queue manager manually through IBM MQ Explorer. The queue manager runs under the logged on user (the interactive user). The queue manager will automatically stop when the interactive user logs off.</p> <p>Select Service (manual) to start the queue manager manually through IBM MQ Explorer. The queue manager runs as a child of the MQ Services service. The queue manager will not automatically stop when the interactive user logs off.</p>	(Not applicable.)
Command server control	<p>To configure the command server so that it starts automatically when the queue manager starts, click Queue Manager; to configure the command server so that it does not start automatically and must be started manually, click Manual.</p>	SCMDSERV

Attribute	Meaning	MQSC parameter
Channel init control	To configure the channel initiator so that it starts automatically when the queue manager starts, click Queue Manager ; to configure the channel initiator so that it does not start automatically and must be started manually, click Manual .	SCHINIT

Extended


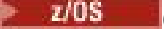


The following table lists the attributes that you can set on the **Extended** page of the Queue Manager properties dialog. The **Default bind type** attribute on the **Extended** page relates to the DefaultBindType stanza key in the configuration files.

Attribute	Meaning	MQSC parameter
Dead-letter queue	Select the name of the queue that the queue manager uses as the dead-letter queue.	DEADQ
Trigger interval	Type the number of milliseconds, from 0 to 999999999, that the queue manager must wait between triggering messages for a queue. This attribute is used only when the Trigger type attribute in the Queue properties is set to First.	TRIGINT
Max uncommitted messages	Type the maximum number of uncommitted messages within a syncpoint, from 1 to 999999999, to limit the number of messages that can be retrieved and put within any single syncpoint. This attribute is not used for messages that are put or retrieved outside syncpoint.	MAXUMSGS
Max handles	Type the maximum number of open handles, from 0 to 999999999, that any one task can have at the same time.	MAXHANDS
Max message length	Type the maximum length of messages, from 32 KB to 100 MB, that is allowed on queues on the queue manager. The default value is 4 MB (4 194 304 bytes). If you reduce the maximum message length for the queue manager, you must also reduce the maximum length of the SYTEM.DEFAULT.LOCAL.QUEUE definition, and all other queues that are connected to the queue manager. This ensures that the queue manager's limit is not less than the limit of any of the queue manager's queues. If you do not do this, and applications inquire only the value of the queue's Max message length attribute, the applications might not work properly.	MAXMSGL
Max properties length	This value controls the size in bytes of the property data that can flow with messages in a V7 queue manager. If the size of the properties exceeds the maximum properties length then the message is rejected.	MAXPROPL
Max priority	Read-only. This is the maximum priority of the queue manager, which is 9.	MAXPRTY

Attribute	Meaning	MQSC parameter
Message mark browse interval	<p>Type the time interval in milliseconds after which the queue manager automatically unmarks browsed messages. The interval can be set to a maximum of 999999999 milliseconds. The interval can also be set to Unlimited. The default value is 5000.</p> <p> Attention: You should not reduce the value below the default of 5000.</p>	MARKINT
Command input queue	Read-only. This is the name of the system-command input queue. Suitably authorized applications can put commands on this queue.	COMMANDQ
Syncpoint	<p>Read-only. This attribute states whether syncpoint is available with the queue manager. Syncpoint is always available on the following platforms:</p> <ul style="list-style-type: none"> •  z/OS •  Windows •   UNIX and Linux 	SYNCPT
Distribution lists	Read-only. This attribute states whether the queue manager supports distribution lists. This attribute is valid only on AIX, HP-UX, IBM i, Solaris, and Windows.	DISTL
UNIX and Linux (only) Application group	The Application group option specifies the application group that clients connecting should belong to. The default is to not belong to an application group.	(Not applicable.)
*Default bind type	This is the default bind type that is used if the application does not specify a bind type on the MQCNO parameter of the MQCONN call. Select SHARED , or ISOLATED .	(Not applicable.)
*Error log size	Specify the size of the queue manager error log at which the log is copied to the backup. The value must be 1048576 - 2147483648 bytes. The default value is 262144 bytes (256 KB).	(Not applicable.)
*Excluded messages	<p>Your IBM MQ system might produce a large number of information messages if the system is heavily used. You can therefore exclude certain messages if required. Type the message ID of each message that is not to be written to the queue manager error log. Enter a comma-separated list of message IDs from the following list:</p> <ul style="list-style-type: none"> • 7163 - Job started message (IBM i only) • 7234 - Number of messages loaded • 9001 - Channel program ended normally • 9002 - Channel program started • 9202 - Remote host not available • 9524 - Remote queue manager unavailable • 9528 - User requested closure of channel • 9999 - Channel program ended abnormally 	(Not applicable.)

Attribute	Meaning	MQSC parameter
*Suppressed messages	<p>Your IBM MQ system might produce a large number of information messages if the system is heavily used. You can prevent selected messages being sent to the console or to the hardcopy log if required. Type the message ID of each message that will be written to the queue manager error log only once in a specified time interval. The time interval is specified in the Suppressed Messages Interval attribute. Enter a comma-separated list of message IDs from the following list:</p> <ul style="list-style-type: none"> • 7163 - Job started message (IBM i only) • 7234 - Number of messages loaded • 9001 - Channel program ended normally • 9002 - Channel program started • 9202 - Remote host not available • 9524 - Remote queue manager unavailable • 9528 - User requested closure of channel • 9999 - Channel program ended abnormally <p>If the same message ID is specified in both the Excluded Messages and the Suppressed Messages attributes, the message is excluded.</p>	(Not applicable.)
*Suppressed messages interval	Type the time interval, in seconds, in which messages that are specified in the Suppressed Messages attribute will be written to the queue manager error log only once. The value must be 1 - 86400 seconds. The default value is 30 seconds.	(Not applicable.)
Custom	<p>The Custom parameter is reserved for the configuration of new features before separate attributes have been introduced. The possible values are a list of zero or more attributes-value pairs, in MQSC-style syntax, separated by at least 1 space.</p> <p>The attribute names and values are case-sensitive, and must be specified in uppercase. The values can contain spaces and parentheses, but must not contain single quotation marks. Examples of valid syntax are:</p> <ul style="list-style-type: none"> • CUSTOM(' ') • CUSTOM('A(B)') • CUSTOM('C(D) E(F)') • CUSTOM('G(5000) H(9.20.4.6(1415))') <p>The queue manager parses the value, but if the string cannot be parsed according to these rules, or if it contains attributes or values that are not recognized, the queue manager ignores the errors.</p>	CUSTOM

Attribute	Meaning	MQSC parameter
Opening shared queues	(z/OS only) When a queue manager makes an MQOPEN call for a shared queue and the queue manager that is specified in the <i>ObjectQmgrName</i> parameter of the MQOPEN call is in the same queue sharing group as the processing queue manager, the Opening shared queues attribute specifies whether the <i>ObjectQmgrName</i> is used or whether the processing queue manager opens the shared queue directly. Use the queue manager specified in ObjectQmgrName means that the <i>ObjectQmgrName</i> is used, and the appropriate transmission queue is opened; Use the local queue manager means that the processing queue manager opens the shared queue directly, which can reduce the traffic in your queue manager network.	SQQMNAME
Intra-group queuing	(z/OS only) Specify whether intra-group queuing is used. To use the shared transmission queue (SYSTEM.QSG.TRANSMIT.QUEUE) when queue managers within a queue sharing group exchange messages, click Enabled ; to use non-shared transmission queues and channels when queue managers within a queue sharing group exchanged messages, click Disabled . If you disable intra-group queuing, the same mechanism for message transfer is used as when the queue managers are not part of a queue sharing group.	IGQ
IGQ user ID	(z/OS only) Specify a user identifier to be used by the IGQ agent to establish authority to put messages to a destination queue. The queue manager must be a member of a queue sharing group to use this attribute. To specify that the user ID of the receiving queue manager within the queue sharing group is to be used as the IGQ user ID, leave the field blank.	IGQUSER
IGQ authority check type	(z/OS only) Specify the type of authority checking and, therefore, the user identifiers, to be used by the IGQ agent. This establishes the authority to put messages to the destination queue. The queue manager must be a member of a queue sharing group to use this attribute. To specify that the default user ID is used to establish authority, click Default ; to specify that the IGQ user ID and the ALT user ID is used to establish authority, click Alternate or IGQ ; to specify that only the IGQ user ID is used to establish authority, click Only IGQ ; to specify that the user ID from the <i>UserIdentifier</i> field in the message descriptor, of a message on the SYSTEM.QSG.TRANSMIT.QUEUE queue, is used to establish authority, click Context .	IGQAUT
Expiry interval	(z/OS only) Type the approximate value in seconds that specifies how often queues are scanned to discard expired messages. the value must be 1 - 99999999. The minimum scan interval is 5 seconds, even if you specify a lesser value 1 - 4. Specify an interval of 0 to ensure that the queues are not scanned; this is the default value.	EXPRYINT

Attribute	Meaning	MQSC parameter
Security profile case	(z/OS only) Specify whether the queue manager supports security profile names in mixed case or in uppercase only. Select Mixed to specify that security names can be in uppercase or in mixed case. Select Upper to specify that security profile names must be in uppercase. This is the default value.	SCYCASE
  Group units of recovery	<p>(z/OS only) This attribute can have a value of either Enabled or Disabled. The value can only be set to Enabled if the system has been configured correctly, otherwise an error is returned. Enabled is the default value.</p> <p>The system programmer must use the code specified to identify which configuration check failed. They must then take corrective action and then reissue the ALTER QMGR command.</p> <p>When you enable group units of recovery (GROUPPUR support) a number of configuration checks are performed to ensure that the configuration steps have been completed. You cannot enable this support if any of these checks fail.</p> <p>These checks are also performed when the queue manager starts if the GROUPPUR 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 GROUPPUR queue manager attribute.</p> <p>For more information, see CSQM507E in the IBM MQ online product documentation.</p>	GROUPPUR
  Loss of coupling facility connectivity	<p>(z/OS only) Specifies the action taken when the queue manager loses connectivity to the administration structure or any CF structures with CFCONLOS set to As queue manager. The two options are:</p> <ul style="list-style-type: none"> • Terminate. This is the default value. The queue manager terminates when connectivity to CF structures is lost. • Tolerate. The queue manager tolerates a loss of connectivity to CF structures and does not terminate. Tolerate can only be set if all queue managers in the queue sharing group are at command level 710 or later. 	CFCONLOS

Exits (Multiplatforms)



The following table lists the attributes that you can set on the **Exits** page of the Queue Manager properties dialog. To configure the queue manager to run user exits, edit the attributes on the **Exits** page. The attributes on the **Exits** page relate to stanzas in the configuration files.

Attribute	Meaning	Stanza key
*Exit default path	If the queue manager is 32 bit, type the path to the location in which exits for this queue manager are stored by default.	ExitsDefaultPath

Attribute	Meaning	Stanza key
*Exit default path (64 bit)	If the queue manager is 64 bit, type the path to the location in which exits for this queue manager are stored by default.	ExitsDefaultPath64
*Local API Exits	Add details of the local API exits that you want to use with this queue manager.	ApiExitLocal
*Name	Specifies the descriptive name of the API exit that is passed to the API exit in the ExitInfoName field of the MQAXP structure. This name must be unique and is limited to a length of 48 characters, and must contain only those characters that are valid for the name of IBM MQ objects, such as queue names.	Name
*Type	Specifies the type of exit: queue manager, or override.	(Not a separate stanza key.)
*Sequence	This attribute is an unsigned numeric value that defines the sequence in which this API exit is called in comparison with other API exits. An API exit with a low sequence number is called before another API exit with a higher sequence number. The order in which different API exits with the same sequence number are called is undefined. It is perfectly valid for there to be gaps in the sequence numbers of the API exits defined for a queue manager.	Sequence
*Module	Specifies the module that contains the code for the API exit. If this field contains the full path name of the module it is used as is. If this field contains just the module name, the module is located using the same method as channel exits; that is, using the value in the Exit default path field on the Exits page of the queue manager properties dialog.	Module
*Function	Specifies the name of the function entry-point into the module that contains the code for the API exit. This entry-point is the MQ_INIT_EXIT function. The length of this field is limited to MQ_EXIT_NAME_LENGTH.	Function

Attribute	Meaning	Stanza key
*Data	If this attribute is specified, leading and trailing blanks are removed, the remaining string is truncated to 32 characters, and the result is passed to the exit in the ExitData field of the MQAXP structure. If this attribute is not specified, the default value of 32 blanks is passed to the exit in the ExitData field of the MQAXP structure.	Data

Cluster

The following table lists the attributes that you can set on the **Cluster** page of the Queue Manager properties dialog. To configure the cluster attributes of the queue manager, edit the attributes on the **Cluster** page.

Attribute	Meaning	MQSC parameter
Cluster membership	Read-only. This table lists the names of the clusters to which the queue manager belongs.	(Not applicable.)
Cluster workload exit	<p>The exit is called when a message is put on a cluster queue. Type the name of the cluster workload exit:</p> <ul style="list-style-type: none"> On UNIX and Linux systems, use the format <code>libraryname(functionname)</code>. The maximum length is 128 characters. On Windows, use the format <code>dllname(functionname)</code>, where <code>dllname</code> is specified without the <code>.dll</code> suffix. The maximum length is 128 characters. On z/OS, type the load module name. The maximum length is 8 characters. On IBM i, use the format <code>progrname libname</code>, where <code>progrname</code> occupies the first 10 characters, and <code>libname</code> occupies the second 10 characters. Add spaces at the end of shorter names to bring them up to 10 characters. The maximum length is 20 characters. 	CLWLEXIT
Cluster workload data	Type the data to be passed to the cluster workload exit when the exit is called. The maximum length of the data is 32 characters.	CLWLDATA

Attribute	Meaning	MQSC parameter
Cluster workload length	<p>Type the maximum number of bytes of message data that is passed to the cluster workload exit:</p> <ul style="list-style-type: none"> • On Windows, type a number from 0 to 104857600 (100 MB). • On other platforms, type a number from 0 to 999999999. 	CLWLLEN
Max outbound cluster channels	<p>Type the maximum number of outbound cluster channels. For more information, see Distributed queuing and clusters in the IBM MQ online product documentation.</p>	CLWLMRUC
Cluster workload mode	<p>The cluster workload exit, CLWL, allows you to specify which cluster queue in the cluster is to be opened in response to an MQI call (for example MQOPEN or MQPUT). The default value is SAFE, which means that the CLWL exit is run in a separate process to the queue manager so that if there is a problem, the integrity of the queue manager is preserved. However, running the CLWL exit as a separate process can have a detrimental effect on performance. To improve performance by running the CLWL exit in the same process as the queue manager, click FAST. Use FAST mode only if you are certain that there are no problems with your CLWL exit because if there is a problem in FAST mode, the queue manager fails and the queue manager's integrity is at risk. The value set for the queue manager will override the value set for the machine-wide configuration.</p>	CLWLMode

Attribute	Meaning	MQSC parameter
CLWL use queue	<p>This attribute specifies whether the queue manager can choose from remote instances of cluster queues as well as local instances. If the queue manager receives a message by way of a cluster channel, the message is put on a local instance of the cluster queue; if the queue manager receives a message locally or by way of a non-cluster channel, and the value of this attribute is Any, the message is put on either local or remote instances of the cluster queue. To enable the queue manager to use remote instances of cluster queues, click Any; to prevent the queue manager using remote instances of cluster queues, click Local. For more information, see Distributed queuing and clusters in the IBM MQ online product documentation.</p>	CLWLUSEQ

Attribute	Meaning	MQSC parameter
Default cluster transmission queue	<p>The default transmission queue type that is used by clustering to transfer messages to other queue managers in the cluster. Messages are transferred by cluster-sender channels.</p> <p>The default value of this attribute is SCTQ. The queue manager uses a single transmission queue to transfer all cluster messages. The transmission queue is SYSTEM.CLUSTER.TRANSMIT.QUEUE.</p> <p>To send messages to each queue manager, in each cluster, using a different transmission queue, set the value of Default cluster transmission queue to Queue for each channel. The queue manager automatically creates a transmission queue if it needs one to send a message to another queue manager in a cluster. The queue is permanent-dynamic. It is created from the model queue SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE. The name of each transmission queue is SYSTEM.CLUSTER.TRANSMIT.ChannelName. ChannelName is the name of the cluster-sender channel that transfers messages from the queue.</p>	DEFCLXQ

Repository

The following table lists the attributes that you can set on the **Repository** page of the Queue Manager properties dialog. To specify that the queue manager hosts the repository for one or more clusters, edit the attributes on the **Repository** page.

Attribute	Meaning	MQSC parameter
Not a cluster full repository	To specify that the queue manager is not a full repository for a cluster, select this option.	(Not applicable.)
Full repository for a cluster	To make this queue manager a full repository for just one cluster, select this option, then type the name of the cluster.	REPOS
Full repository for a list of clusters	To make this queue manager a full repository for more than one cluster, select this option, then type the name of the cluster.	REPOSNL

Communication

The following table lists the attributes that you can set on the **Communication** page of the Queue Manager properties dialog. To configure how the queue manager sends and receives messages, edit the attributes on the **Communication** page.

Attribute	Meaning	MQSC parameter
Default transmission queue	Type the name of the default transmission queue on which messages that are destined for a remote queue manager are put if there is no other suitable transmission queue defined. The queue named must be a local transmission queue, but not the cluster transmission queue.	DEFXMITQ
Channel auto definition	To enable receiver- and server-connection channels to be defined automatically, click Enabled ; to prevent receiver- and server-connection channels being defined automatically, click Disabled . Cluster-sender channels can always be defined automatically, regardless of the setting of this attribute.	CHAD

Attribute	Meaning	MQSC parameter
Channel auto definition exit	<p>The exit is called when an inbound request for an undefined receiver, server-connection, or cluster-sender channel is received. The exit is also called when starting a cluster-receiver channel. Type the name of the channel auto-definition exit:</p> <ul style="list-style-type: none"> • On Windows, use the format <code>dllname(functionname)</code>, where <code>dllname</code> is specified with the <code>.dll</code> suffix. The maximum length is 128 characters. • On IBM i, use the format <code>progrname libname</code>, where <code>progrname</code> occupies the first 10 characters and <code>libname</code> occupies the second 10 characters. Add spaces at the end of shorter names to bring them up to 10 characters. The maximum length is 20 characters. • On UNIX and Linux, use the format <code>libraryname(functionname)</code>. The maximum length is 128 characters. • On z/OS, type the load module name. The maximum length is eight characters. 	CHADEXIT
Channel authentication	<p>To exercise more precise control over the access granted to connecting systems at a channel level, you can use channel authentication records. IBM WebSphere MQ 7.1, and later, queue managers are created using channel authentication by default. If you migrate a queue manager to IBM WebSphere MQ 7.1 from an earlier release, channel authentication is not enabled. To enable channel authentication, use the command <code>ALTER QMGR CHLAUTH(ENABLED)</code></p>	CHLAUTH

Attribute	Meaning	MQSC parameter
Reverse lookup of host name	<p>Controls whether reverse lookup of the host name from a Domain Name Server (DNS) is done for the IP address from which a channel has connected. This attribute only has an effect on channels using a transport type (TRPTYPE) of TCP.</p> <p>If you are using channel authentication rules with CHLAUTH(ENABLED) and you have defined any rules that use a DNS host name in the ADDRESS field of the rule, these rules will never match an inbound channel if REVDNS is set to DISABLED.</p> <p>Changes to this parameter take effect the next time a channel starts up. Channels that have already obtained host name information by reverse looking up an IP address retain this information.</p>	REVDNS
IP address version	To specify that the queue manager uses the IPv6 protocol, click IPV6 ; to specify that the queue manager uses the IPv4 protocol, click IPV4 .	IPADDRV

Attribute	Meaning	MQSC parameter
Activity recording	<p>If a queue manager application performs some work on behalf of a message in which activity reports were requested, the queue manager can generate an activity report. You can use this activity report to work out where the message went in the queue manager network. To prevent queue manager applications generating activity reports, click Disabled; to enable queue manager applications to generate activity reports, click Message or Queue. If you click Message, a queue manager application that generates an activity report puts the report on the queue that the originator of the message requested in the ReplyToQ and ReplyToQMgr fields of the message descriptor; if you click Queue, a queue manager application that generates an activity report puts the report on the system queue SYSTEM.ADMIN.ACTIVITY.QUEUE. If you make changes to this attribute, you must stop and restart any channels to which you want the changes to apply.</p>	ACTIVREC

Attribute	Meaning	MQSC parameter
Trace-Route recording	<p>You can use Trace-Route messages to determine the routes that messages take through a queue manager network. Participating queue manager applications can generate information about the route and put the information in activity reports. The queue manager applications can also add the information to the Trace-Route message itself, depending on the options set in the Trace-Route message. The Trace-Route message can then accumulate chronological information about the route. The Activity recording attribute specifies on which queue the activity reports are put. The Trace-route recording attribute controls the accumulation of information in the Trace-Route message itself. To prevent the queue manager appending route information within the Trace-Route message and from returning information in reply messages, click Disabled. To enable queue manager applications to add the route information to the Trace-Route message, click Message or Queue. If you click Message and a queue manager application generates a reply message containing the accumulated route information from the Trace-Route message, the queue manager application puts the reply message on the queue that the originator of the message requested in the ReplyToQ and ReplyToQMGr fields of the message descriptor; if you click Queue and a queue manager application generates a reply message containing the accumulated route information from the Trace-Route message, the queue manager application puts the reply message on the system queue</p> <p>SYSTEM.ADMIN.TRACE.ROUTE.QUEUE. If you make changes to this attribute, you must stop and restart any channels to which you want the changes to apply.</p>	ROUTEREC

Events

The following table lists the attributes that you can set on the **Events** page of the Queue Manager properties dialog. To configure the queue manager to generate events in response to certain criteria, edit the attributes on the **Events** page.



Attribute	Meaning	MQSC parameter
Authority events	When an application tries to open a queue for which the application does not have the required authority, the queue manager can generate an authorization event message. To generate authorization event messages, click Enabled ; to prevent the queue manager generating authorization event messages, click Disabled .	AUTHOREV
Inhibit events	When an application tries to put a message on a queue that is put-inhibited, or to get a message from a queue that is get-inhibited, the queue manager can generate an inhibit event message. To generate inhibit event messages, click Enabled ; to prevent the queue manager generating inhibit event messages, click Disabled .	INHIBTEV
Local events	When an application or the queue manager has not been able to access an object, for example because the object has not been defined, the queue manager can generate a local event message. To generate local event messages, click Enabled ; to prevent the queue manager generating local event messages, click Disabled .	LOCALEV
Remote events	When an application or the queue manager cannot access a queue on another queue manager, for example the transmission queue is not correctly defined, the queue manager can generate a remote event message. To generate remote event messages, click Enabled ; to prevent the queue manager generating remote event message, click Disabled .	REMOTEEV

Attribute	Meaning	MQSC parameter
Start and stop events	When a queue manager starts, or has been requested to stop or quiesce (z/OS supports only start), the queue manager can generate a start and stop event message. To generate start and stop event messages, click Enabled ; to prevent the queue manager generating start and stop messages, click Disabled .	STRSTPEV
Performance events	When a resource reaches a threshold condition, for example a queue depth limit has been reached, the queue manager can generate a performance event message. To generate performance event messages, click Enabled ; to prevent the queue manager generating performance event messages, click Disabled .	PERFMEV
Command events	When an MQSC command or PCF command is executed successfully, the queue manager can generate command event messages. To generate command event messages, click Enabled ; to prevent the queue manager generating command events, click Disabled ; to generate command event messages except DISPLAY MQSC commands and Inquire PCF commands, click No Display .	CMDEV
Channel events	When the queue manager detects certain conditions on a channel, for example the channel starts or stops, the queue manager can generate channel event messages. To generate channel event messages, click Enabled ; to prevent the queue manager generating channel event messages, click Disabled .	CHLEV
Channel auto definition events	When a channel is automatically generated, the queue manager can generate a channel auto-definition event message. To generate channel auto-definition event messages, click Enabled ; to prevent the queue manager generating channel auto-definition event messages, click Disabled .	CHADEV

Attribute	Meaning	MQSC parameter
SSL events	When a channel that uses TLS security fails to establish an TLS connection, the queue manager can generate an SSL event message. To generate SSL event messages, click Enabled ; to prevent the queue manager generating SSL event messages, click Disabled .	SSLEV
Configuration events	When an object is created or modified, the queue manager can generate a configuration event message. To generate configuration event messages, click Enabled ; to prevent the queue manager generating configuration event messages, click Disabled .	CONFIGEV
Bridge events	(z/OS only) When an IMS bridge starts or stops, the queue manager can generate a bridge event message. To generate bridge event messages, click Enabled ; to prevent the queue manager generating bridge event messages, click Disabled .	BRIDGEEV
Logger events	When a queue manager is configured to use linear logging, the queue manager can be configured to generate a logger event message when changes are written to the IBM MQ recovery log. To generate logger event messages, click Enabled ; to prevent the queue manager generating logger event messages, click Disabled .	LOGGEREV

SSL

The following table lists the attributes that you can set on the **SSL** page of the Queue Manager properties dialog. To configure the queue manager and its channels to use TLS security, edit the attributes on the **SSL** page.

Attribute	Meaning	MQSC parameter
Key repository	Type the full path to the key repository for the queue manager.	SSLKEYR
Certificate label		CERTLABL
  Queue sharing group certificate label		CERTQSGL

Attribute	Meaning	MQSC parameter
Revocation namelist	<p>Type the name of the revocation namelist. The revocation namelist can contain a mixture of authentication information objects from either, or both, of the following types:</p> <ul style="list-style-type: none"> • CRL LDAP authentication information objects, which store connection information for LDAP servers that contain Certification Revocation Lists (CRLs). • Online Certificate Status Protocol (OCSP) authentication information objects, which store connection information for OCSP responders. 	SSLCRLNL
Cryptographic hardware	To configure your cryptographic hardware, click Configure in the Cryptographic hardware settings dialog, enter the details of your cryptographic hardware.	SSLCRYP
SSL Reset Count	Type the number of unencrypted bytes, from 0 to 999999999, that are sent and received within an TLS conversation before the secret key is renegotiated. A value of 0 means that the secret key is never renegotiated. The number of bytes includes control information that is sent by the message channel agent (MCA). If the value of this attribute is greater than 0 and the value of the <u>Heartbeat interval</u> attribute in the <u>Channel properties</u> is greater than 0, the secret key is also renegotiated before message data is sent or received following a channel heartbeat.	SSLRKEYC
SSL FIPS required	To specify whether only FIPS-certified cryptographic algorithms are to be used (if the cryptography is performed in IBM MQ rather than cryptographic hardware), click Yes . To specify that any cryptographic algorithm can be used, click No .	SSLFIPS

Attribute	Meaning	MQSC parameter
OCSP authentication	<p>The OCSP authentication setting dictates the outcome of a connection in the event of an 'Unknown' response from the OCSP call.</p> <ul style="list-style-type: none"> • Required: IBM MQ rejects the connection. • Optional: The connection is allowed to succeed. • Warn: The connection is also allowed to succeed and IBM MQ issues a message of type AMQ9717 into the error logs. 	N/A
OCSP check extensions	<p>The OCSP check extensions attribute controls whether the OCSP server details in AuthorityInfoAccess certificate extensions, are used to perform a digital revocation check. There are 2 possible values for the attribute:</p> <ul style="list-style-type: none"> • Yes: A digital certificate revocation check is performed. This is the default value. • No: A digital certificate revocation check is not performed. 	N/A
SSL HTTP proxy name	<p>The SSL HTTP proxy name is either the host name or network address of the HTTP proxy server which is to be used by GSKit for OCSP checks. This address can optionally be followed by a port number, enclosed in parentheses. If you do not specify the port number, the default HTTP port, 80, is used.</p>	N/A
Suite B strength	<p>The Suite B strength attribute controls whether Suite B encryption is used. There are four possible values for the attribute:</p> <ul style="list-style-type: none"> • 128-bit • 192-bit • none • 128-bit and 192-bit 	SUITEB

Attribute	Meaning	MQSC parameter
Certificate validation policy	<p>The certificate validation policy attribute controls which TLS certificate validation policy is used to validate digital certificates received from remote partners. There are two possible values for the attribute:</p> <ul style="list-style-type: none"> • ANY • RFC5280 <p>Changes to this attribute take effect only after a refresh security command has been issued. For information about how to refresh security in the IBM MQ Explorer, see “Refreshing TLS security” on page 200.</p>	CERTVPOL

Statistics

The following table lists the attributes on the **Statistics** page of the Queue Manager properties dialog. The **Statistics** page displays the information about the history of the queue manager. You cannot edit any of these attributes.

Attribute	Meaning	MQSC parameter
Creation date	Read-only. This is the date on which the queue was created.	CRDATE
Creation time	Read-only. This is the time at which the queue was created.	CRTIME
Alteration date	Read-only. This is the date on which the queue's attributes were last altered.	ALTDATE
Alteration time	Read-only. This is the time at which the queue's attributes were last altered.	ALTTIME
QMID	Read-only. This is the internally-generated unique name of the queue manager.	QMID

Online monitoring

The following table lists the attributes that you can set on the **Online monitoring** page of the Queue Manager properties dialog. To collect data about the current performance of the queue manager's channels and queues, edit the attributes on the **Online monitoring** page.

Attribute	Meaning	MQSC parameter
Channel monitoring	<p>This attribute specifies whether to collect online monitoring data about the current performance of channels hosted by the queue manager. To switch off online monitoring data collection for the queue manager's channels that have the value Queue Manager in their Channel monitoring attribute, click Off; to switch off online monitoring data collection for all the queue manager's channels regardless of the setting of the channel's Channel monitoring attribute, click None. To specify a low rate of data collection, with a minimal effect on system performance, for channels that have the value Queue Manager in their Channel monitoring attribute, click Low; to specify a moderate rate of data collection, with limited effect on system performance, for channels that have the value Queue Manager in their Channel monitoring attribute, click Medium; to specify a high rate of data collection, with a likely effect on system performance, for channels that have the value Queue Manager in their Channel monitoring attribute, click High.</p>	MONCHL

Attribute	Meaning	MQSC parameter
Queue monitoring	<p>This attribute specifies whether to collect online monitoring data about the current performance of queues hosted by the queue manager. To switch off online monitoring data collection for the queue manager's queues that have the value Queue Manager in their Queue monitoring attribute, click Off; to switch off online monitoring data collection for all the queue manager's queues regardless of the setting of the queue's Queue monitoring attribute, click None. To specify a low rate of data collection, with a minimal effect on system performance, for queues that have the value Queue Manager in their Queue monitoring attribute, click Low; to specify a moderate rate of data collection, with limited effect on system performance, for queues that have the value Queue Manager in their Queue monitoring attribute, click Medium; to specify a high rate of data collection, with a likely effect on system performance, for queues that have the value Queue Manager in their Queue monitoring attribute, click High.</p>	MONQ

Attribute	Meaning	MQSC parameter
Auto CLUSSDR monitoring	<p>This attribute specifies whether to collect online monitoring data about the current performance of auto-defined cluster-sender channels. To inherit from the value of the queue manager's Channel monitoring attribute, click Queue Manager; to switch off data collection for auto-defined cluster-sender channels on the queue manager, click None; to specify a low rate of data collection with a minimal effect on system performance, click Low (the data that is collected is unlikely to be the most current); to specify a moderate rate of data collection with limited effect on system performance, click Medium; to specify a high rate of data collection with a likely effect on system performance, click High (the data that is collected is the most current available).</p>	MONACLS
Activity trace override	<p>This attribute specifies whether applications can override the value of the queue manager attribute ACTVTRC. Valid values are Enabled and Disabled. When Enabled is selected, applications can override the settings of the ACTVTRC parameter by using the options field of the MQCNO structure of the MQCONN API call. When Disabled is selected, applications cannot override the settings of the ACTVTRC parameter. Disabled is the default value for this parameter. Changes to this parameter are effective for connections to the queue manager that occur after the change. This parameter is valid only on IBM i, UNIX and Windows systems.</p>	ACTVCONO

Attribute	Meaning	MQSC parameter
Activity trace	This attribute specifies whether MQI application activity tracing information is to be collected. Valid values are On and Off . When On is selected, MQI application activity tracing information collection is enabled. If queue manager attribute ACTVCONO is set to Enabled , the value of this parameter can be overridden using the options field of the MQCNO structure. When Off is selected, MQI application activity tracing information collection is disabled. Off is the default value for this parameter. Changes to this parameter are effective for connections to the queue manager that occur after the change. This parameter is valid only on IBM i, UNIX and Windows systems.	ACTVTRC

Statistics monitoring (Multiplatforms)



The following table lists the attributes that you can set on the **Statistics monitoring** page of the Queue Manager properties dialog. To collect statistical data on the activity of the queue manager, edit the attributes on the **Statistics monitoring** page. For z/OS Statistics monitoring settings, see “Statistics monitoring (z/OS)” on page 403.

Attribute	Meaning	MQSC parameter
MQI Statistics	To collect MQI statistics data for the queue manager, click On ; to prevent MQI statistics monitoring data collection for the queue manager, click Off .	STATMQI
Queue Statistics	This attribute specifies whether to collect statistics data about the activity of queues hosted by the queue manager. To switch on statistics data collection for the queue manager's queues, click On ; to switch off statistics data collection for the queue manager's queues that have the value Queue Manager in their Queue statistics attribute (see Queue properties), click Off ; to switch off statistics data collection for all the queue manager's queues regardless of the setting of the queue's Queue statistics attribute, click None .	STATQ

Attribute	Meaning	MQSC parameter
Channel Statistics	<p>This attribute specifies whether to collect statistics data about the activity of channels hosted by the queue manager. To switch off statistics data collection for the queue manager's channels that have the value <code>Queue Manager</code> in their <code>Channel statistics</code> attribute (see Channel properties), click Off; to switch off statistics data collection for all the queue manager's channels regardless of the setting of the channel's <code>Channel statistics</code> attribute, click None. To specify a low rate of data collection, with a minimal effect on system performance, for channels that have the value <code>Queue Manager</code> in their <code>Channel statistics</code> attribute, click Low; to specify a moderate rate of data collection, with limited effect on system performance, for channels that have the value <code>Queue Manager</code> in their <code>Channel statistics</code> attribute, click Medium; to specify a high rate of data collection, with a likely effect on system performance, for channels that have the value <code>Queue Manager</code> in their <code>Channel statistics</code> attribute, click High.</p>	STATCHL
Auto CLUSSDR Statistics	<p>This attribute specifies whether to collect statistics data about the activity of auto-defined cluster-sender channels. To inherit from the value of the queue manager's <code>Channel statistics</code> attribute, click Queue Manager; to switch off data collection for auto-defined cluster-sender channels on the queue manager, click None; to specify a low rate of data collection, click Low (the data that is collected is unlikely to be the most current); to specify a moderate rate of data collection, click Medium; to specify a high rate of data collection, click High (the data that is collected is the most current available).</p>	STATACLS

Attribute	Meaning	MQSC parameter
Statistics Interval	Type the interval, in seconds, between writing the statistics monitoring data to the monitoring queue. The default value is 1800 seconds (30 minutes).	STATINT

Accounting monitoring (Multiplatforms)



The following table lists the attributes that you can set on the **Accounting monitoring** page of the Queue Manager properties dialog. To collect data about the activity of a connection, edit the attributes on the **Accounting monitoring** page. For z/OS Account monitoring settings, see [Accounting monitoring on z/OS](#).

Attribute	Meaning	MQSC parameter
MQI accounting	To collect MQI accounting data for the queue manager, click On ; to prevent MQI accounting monitoring data collection for the queue manager, click Off .	ACCTMQI
Queue accounting	This attribute specifies whether to collect accounting data about the activity of connections for queues hosted by the queue manager. To enable accounting data collection for the queue manager's queues, click On ; to disable accounting data collection for the queue manager's queues that have the value Queue Manager in their Queue accounting attribute (see Queue properties), click Off ; to disable accounting data collection for all the queue manager's queues regardless of the setting of the queue's Queue accounting attribute, click None .	ACCTQ
Accounting interval	Type the interval, in seconds, between writing the accounting monitoring data to the monitoring queue. The default value is 1800 seconds (30 minutes).	ACCTINT
Accounting conn override	Applications can override the MQI accounting attribute and the Queue accounting attribute using the Connect options in MQCONN calls. To enable applications to override the attributes, click Enabled ; to prevent applications overriding the attributes, click Disabled .	ACCTCONO



Log (Multiplatforms)



The following table lists the attributes that you can set on the **Log** page of the Queue Manager properties dialog. To configure the log settings for the queue manager, edit the attributes on the **Log** page. The attributes on the **Log** page relate to stanzas in the configuration files. For z/OS log settings, see [Log \(z/OS only\)](#).

Attribute	Meaning	Stanza key
*Log type	Read-only. This attribute shows the type of logging that the queue manager uses. You cannot change the type of logging after the queue manager is created.	LogType
*Log path	Read-only. This attribute shows the location of the queue manager's logs. You cannot change the value of the Log path attribute after the queue manager is created.	LogDefaultPath
*Log file pages	Read-only. This attribute shows the number of 4 KB pages in the log file. For example, if the value is 256, the file size is 1 MB. The default value is 4096, the file size 16 MB.	LogFileSize
*Log primary files	These are the log files that are allocated when the queue manager is created. On Windows, type the number, from 2 to 254, of primary log files. The default value is 3. The total number of primary and secondary log files must not exceed 255 and must not be less than 3. On UNIX and Linux, type the number, from 2 to 510, of primary log files. The default value is 3. The total number of primary and secondary log files must not exceed 511 and must not be less than 3. The value is examined when the queue manager is created or started. You can change the value after the queue manager has been created but the change is not effective until the queue manager is restarted.	LogPrimaryFiles

Attribute	Meaning	Stanza key
*Log secondary files	<p>These are the log files that are allocated when the primary files are exhausted.</p> <p>On Windows, type the number, from 1 to 253, of secondary log files. The default value is 3. The total number of primary and secondary log files must not exceed 255 and must not be less than 3.</p> <p>On UNIX and Linux, type the number, from 1 to 509, of secondary log files. The default value is 3. The total number of primary and secondary log files must not exceed 511 and must not be less than 3.</p>	LogSecondaryFiles
*Log buffer pages	<p>Type the number, from 0 to 4096, of 4 KB buffer pages for writing. If you type a number from 1 to 17, the minimum of 18 (72 KB) is used. If you type a number from 18 to 4096, that number of pages is used. If you type 0, the queue manager selects the size. In IBM WebSphere MQ for Windows 7.0 and later, this is 512 (2048 KB).</p> <p>The default value is 0 (Which the queue manager selects as 512 (2048 KB). If you change the value of this property, restart the queue manager to apply the change.</p>	LogBufferPages
*Log write integrity	<p>This is the method that the logger uses to reliably write log records. If you are using a non-volatile write cache (for example, ssa write cache enabled), it is safe for the logger to write log records in a single write, so click SingleWrite; if you need to write log records with more integrity, click DoubleWrite to use an additional write; if you need to write log records with complete integrity but at the cost of performance, click TripleWrite to use another additional write.</p>	LogWriteIntegrity

Attribute	Meaning	Stanza key
  Log management	<p>The method used to manage your logs. LogManagement applies only when LogType is LINEAR.</p> <p>If you change the LogManagement value, the change does not take effect until the queue manager is restarted.</p> <p>There are three options.</p> <p>Manual, where you manage the log extents manually. Specifying this option means that the queue manager does not reuse or delete log extents, even when they are no longer required for recovery.</p> <p>Automatic, where log extents are managed automatically by the queue manager. Specifying this option means that the queue manager is able to reuse or delete log extents as soon as they are no longer required for recovery. No allowance is made for archiving.</p> <p>Archive, where log extents are managed by the queue manager, but you must notify the queue manager when archiving of each log extent is complete.</p> <p>Specifying this option means that the queue manager is free to reuse or delete a log extent, as soon as the queue manager has been notified that an extent no longer required for recovery has been archived.</p> <p>The default value is Manual.</p>	LogManagement

XA resource managers (Multiplatforms)



The following table lists the attributes that you can set on the **XA resource manager** page of the Queue Manager properties dialog. The **XA resource manager** page displays attributes to edit if the queue manager coordinates its own units of work along with database updates; for example, the name of the resource manager (the database) and the location of the switch file, which helps IBM MQ communicate with the database. The attributes on the **XA resource manager** page relate to the XAResourceManager stanza in the configuration files.

Attribute	Meaning	Stanza key
*Name	Type the name of the resource manager (the database).	Name

Attribute	Meaning	Stanza key
*SwitchFile	Type the location of the switch file, which helps IBM MQ to communicate with the database.	SwitchFile
*XAOpenString	You can type a string of data that IBM MQ passes in its calls to the database manager's xa_open function. IBM MQ and the queue manager call the xa_open function when the queue manager starts and when you make the first call to MQBEGIN in your IBM MQ application process. The default is a zero-length string.	XAOpenString
*XACloseString	You can type a string of data that IBM MQ passes in its calls to the database manager's xa_close function. IBM MQ and the queue manager call the xa_close function when the queue manager starts and when you make a call to MQDISC in your IBM MQ application process, having earlier made a call to MQBEGIN. The default is a zero-length string. It is common to have a zero-length string.	XACloseString
*ThreadOfControl	The queue manager uses this value for serialization purposes. If the database client allows threads to call the XA functions without serialization, the value for ThreadOfControl can be THREAD. If the database client does not allow threads to call its XA functions in this way, the value for ThreadOfControl must be PROCESS. The default is PROCESS.	ThreadOfControl

Installable services (Multiplatforms)



The following table lists the attributes on the **Installable services** page of the Queue Manager properties dialog. The **Installable services** page displays information about the installable services installed on your computer. By default, only the authorization service, OAM, is shown. The attributes on the **Installable services** page relate to the Service stanza in the configuration files. For more information, see [Configuring services and components](#) in the IBM MQ online product documentation.

Attribute	Meaning	Stanza key
*Service name	Read-only. This is the name of the service.	Name

Attribute	Meaning	Stanza key
*Service entry points	Read-only. This is the number of entry points that are defined for the service, including the initialization and termination entry points.	EntryPoints
*Security policy	Read-only. This is the security policy for the queue manager. Default means that the default security policy is used; NTSDs Required means that a Windows security identifier is passed to the OAM when performing security checks.	SecurityPolicy
*ServiceComponents	Read-only. This is a list of the service components installed on your computer.	ServiceComponents
*Name	Read-only. This is the name of the component.	component_name
*Service	Read-only. This is the name of the installable service.	service_name
*Data size	Read-only. This is the size, in bytes, of the component data area passed to the component on each call. A value of 0 is used if no component data is required.	size
*Module	Read-only. This is the path to the module that contains the code for the component.	module_name

Channels

The following table lists the attributes that you can set on the **Channels** page of the Queue Manager properties dialog. To configure the behavior of the queue manager's channels, edit the attributes on the **Channels** page.

Channel attributes for Multiplatforms

On Multiplatforms, the attributes on the **Channels** page relate to stanzas in the configuration files. You cannot edit this attributes on remote distributed queue managers.

Attribute	Meaning	Stanza key
*Max channels	Type the maximum number of channels, from 1 to 9 999, that can be current (including server-connection channels with connected clients). For z/OS, the value must be between 1 and 9999, with a default value of 200. For all other platforms, the value must be between 1 and 65535, with a default value of 100. On a production system, you could use, for example, the value 1000. If the value of this attribute is reduced, any current channels that exceed the new limit continue to run until they stop.	MaxChannels
*Max active channels	Type the maximum number of channels that can be active at any one time. The default is the value specified for the MaxChannels attribute. For z/OS, the value must be between 1 and 9999. For all other platforms, the value must be between 1 and 65535.	MaxActiveChannels
*Max initiators	Type the maximum number of initiators allowed. The default and maximum value is 3.	MaxInitiators
*MQI bind type	Select the type of connection that channels use to connect to applications. To connect using a standard connection, click STANDARD ; to connect without using an agent process, click FASTPATH .	MQBindType
*Adopt new MCA	<p>This attribute specifies whether an orphaned MCA instance is adopted (restarted) when a new inbound channel request is detected that matches the value of the Adopt new MCA check attribute.</p> <p>To adopt all channel types, type All. If a FASTPATH channel cannot be safely ended, it is not ended and the adoption fails.</p> <p>If you do not require orphaned channels to be adopted, type No.</p>	AdoptNewMCAType

Attribute	Meaning	Stanza key
*Adopt new MCA check	<p>This attribute specifies which elements are checked to determine whether an MCA should be adopted when a new inbound channel is detected with the same name as an already active MCA. Type one or more of the following values separated by commas:</p> <ul style="list-style-type: none"> • To check the queue manager name and the network address to prevent your channels from being inadvertently shut down, type ALL • To check the network address, type ADDRESS • To check the queue manager name, type NAME • To check the user ID that the queue manager is running under, type QM • To do no checking, type NONE 	AdoptNewMCACheck
*Adopt new MCA timeout	Type the number of seconds, from 1 to 3600, that the new process must wait for the old process to end. The default value is 60.	AdoptNewMCATimeout
*Pipeline length	To allow an MCA to transfer messages using multiple threads, type the number of concurrent threads that the channel will use. The default is 1; if you type a value greater than 1, it is treated as 2. Make sure that you configure the queue manager at both ends of the channel to have a Pipeline length that is greater than 1. Pipelining is effective only for TCP/IP channels.	PipeLineLength

Channel attributes for z/OS

On z/OS, the channel attributes are not configuration attributes; they are just ordinary queue manager attributes like all the other attributes in the z/OS queue manager properties dialog.

Attribute	Meaning	MQSC parameter
Active channels	Type the maximum number of channels, from 1 to 9999, that can be active at any one time. The default value is 200. The value must not be greater than the value of MAXCHL that defines the maximum number of channels allowed.	ACTCHL

Attribute	Meaning	MQSC parameter
Channel initiator adapters	Specify the number of adapter subtasks, from 0 to 9999, to use for processing IBM MQ calls. As a guide, the ratio of adapters to dispatchers should be approximately 8 to 5. However, if you have only a small number of channels, you do not have to decrease the value of this parameter from the default value. Suggested settings: 8 (default) for a test system; 20 for a production system. Specifying a value of 20 adapters gives greater parallelism of IBM MQ calls. This is important for persistent messages. Specifying fewer adapters might be better for nonpersistent messages.	CHIADAPS
Adopt new MCA check	This attribute specifies which elements are checked to determine whether an MCA should be adopted when a new inbound channel is detected with the same name as an already active MCA. Enter one or more of the following values separated by commas. To check the queue manager name and the network address to prevent your channels from being inadvertently shut down, click ALL ; to check the network address, click Network address ; to check the queue manager name, click Queue manager name ; to do no checking, click NONE .	ADOPTCHK
Adopt new MCA type	This attribute specifies whether an orphaned instance of an MCA of a particular channel type should be restarted automatically when a new inbound channel request matching the adopt new MCA check parameters is detected. This attribute is read-only.	ADOPTTYPE

Attribute	Meaning	MQSC parameter
Channel initiator dispatchers	Specify the number of dispatchers to use for the channel initiator, from 1 to 9999. As a guideline, allow one dispatcher for every 50 current channels. If you have a small number of channels, however, use the default value of 5. If you are using TCP/IP, the maximum number of dispatchers used for TCP/IP is 100, even if you specify a large value for this attribute. You are recommended to specify a value of 20 on production systems in order to handle up to 1000 active channels. Restart the channel initiator to apply changes to this attribute.	CHIDISPS
Register with WLM	This parameter is no longer used. It must have the value No .	DNSWLM
Listener timer	Specify the time interval, in seconds from 5 to 9999, between attempts by IBM MQ to restart the listener after an APPC or TCP/IP failure. When the listener is restarted on TCP/IP, it uses the same port and IP address as it used when it first started. Changes to this attribute take effect for listeners that are subsequently started. Listeners that are currently started are unaffected by changes to this attribute.	LSTRTMR
Channels using LU6.2	Specify the maximum number of channels that can be current, or clients that can be connected, that use the LU 6.2 transmission protocol. Type a value from 0 to 9999. If you type 0, the LU 6.2 transmission protocol is not used. The value must not be greater than the value of MAXCHL that defines the maximum number of channels allowed.	LU62CHL

Attribute	Meaning	MQSC parameter
Maximum channels	Type the maximum number of channels, from 0 to 9999, that can be current (including server-connection channels with connected clients). The default value is 200. On a production system, you could use, for example, the value 1000. If the value of this attribute is reduced, any current channels that exceed the new limit continue to run until they stop. The values of ACTCHL, LU62CHL, and TCPCHL must not be greater than the maximum number of channels.	MAXCHL
Lowest port address	Type the lowest port number, from 0 to 65535, to be used when binding outgoing channels. When all the port numbers between the value of the Lowest port address attribute and the Highest port address attribute have been used, outgoing channel bind to any available port number. The default is 0, which means that all outgoing channels bind to any available port number. Changes to this attribute take effect for channels that are subsequently started. Channels that are currently running are unaffected by changes to this attribute.	OPORTMIN
Highest port address	Type the highest port number, from 0 to 65535, to be used when binding outgoing channels. When all the port numbers between the value of the Lowest port address attribute and the Highest port address attribute have been used, outgoing channel bind to any available port number. The default is 0, which means that all outgoing channels bind to any available port number. Changes to this attribute take effect for channels that are subsequently started. Channels that are currently running are unaffected by changes to this attribute.	OPORTMAX

Attribute	Meaning	MQSC parameter
Receive timeout	Specify the approximate length of time that a TCP/IP channel waits to receive data, including heartbeats, from its partner before returning to the inactive state. This attribute applies only to message channels, and not MQI channels. See also the Receive timeout type attribute for more information.	RCVTIME
Receive timeout type	Set this attribute to specify how the value of the Receive timeout attribute is interpreted. To specify that the Receive timeout value is a multiplier to be applied to the negotiated Heartbeat interval value to determine how long a channel is to wait, set the Receive timeout type attribute to Multiply , then specify a Receive timeout value of zero or in the range 2 through 99 (if you specify zero, the channel does not time out its wait to receive data from its partner); to specify that Receive timeout value is the number of seconds to be added to the negotiated Heartbeat interval value to determine how long a channel is to wait, set Receive timeout type to Add , then specify a Receive timeout value in the range 1 through 999999; to specify that the Receive timeout value is the number of seconds that the channel is to wait, set Receive timeout type to Equal , then specify an Receive timeout value in the range zero through 999999 (if you specify zero, the channel does not time out its wait to receive data from its partner).	RCVTTYPE

Attribute	Meaning	MQSC parameter
Minimum receive timeout	Type the minimum length of time, in seconds from 0 to 999999, that a TCP/IP channel waits to receive data, including heartbeats, from its partner before returning to an inactive state. This attribute applies only to message channels, and not to MQI channels. If you use the <code>Receive timeout</code> type attribute to specify that the TCP/IP channel wait time is relative to the negotiated value of the channel's <code>Heartbeat interval</code> value and the resultant value is less than the value of this attribute, the value of this attribute is used.	RCVTMIN
Channels using TCP	Type the maximum number of channels that can be current, or clients that can be connected, that use the TCP/IP transmission protocol. Type a value from 0 to 9 999 (although note that TCP/IP might not support as many as 9 999 channels). If you type 0, the TCP/IP transmission protocol is not used. The value must not be greater than the value of MAXCHL that defines the maximum number of channels allowed.	TCPCHL
TCP keepalive	Specify whether the Keepalive facility is used to check that the other end of the connection is still available. If it is not available, the channel is closed. To specify that the Keepalive facility is not to be used, click No ; to specify that the Keepalive facility is to be used as specified in the TCP profile configuration data set, click Yes (the interval is specified in the <code>Keepalive Interval</code> attribute on the Extended page of the Channel properties).	TCPKEEP
TCP stack type	To specify that the channel initiator uses only the TCP/IP address space that is specified in the <code>TCP name</code> attribute, click Single ; to specify that the channel initiator can use multiple TCP/IP address spaces, and the default value is the value of the <code>TCP name</code> attribute, click Multiple .	TCPSTACK

Attribute	Meaning	MQSC parameter
Channel initiator trace auto start	To specify that the channel initiator trace starts automatically, click Yes ; to specify that the channel initiator trace does not start automatically, click No . Changes to this parameter take effect when the channel initiator is restarted. If you need to start or stop channel initiator trace without restarting the channel initiator, use the Trace dialog after the channel initiator has started. To open the Trace dialog, in the Navigator view, right-click IBM MQ , then click Trace	TRAXSTR
Channel initiator trace table size	Type the size, in megabytes from 2 to 2048, of the channel initiator's trace data space. Note that changes to this attribute take effect immediately; any existing trace table contents are lost. When you use large z/OS data spaces, ensure that sufficient auxiliary storage is available on your system to support any related z/OS paging activity. You might also need to increase the size of your SYS1.DUMP data sets.	TRAXTBL
DNS group name	This parameter is no longer used.	DNSGROUP
LU group name	Type the generic LU name to be used by the LU 6.2 listener that handles inbound transmissions for the queue sharing group. The maximum length of this parameter is 8 characters. If this name is blank, the listener cannot be used. Changes to this attribute take effect for listeners that are subsequently started. Listeners that are currently started are unaffected by changes to this attribute.	LUGROUP
LU name	Type the name of the LU to use for outbound LU 6.2 transmissions. Set this to be the same as the name of the LU to be used by the listener for inbound transmissions. The maximum length of this parameter is 8 characters. If this name is blank, the APPC/MVS default LU name should be used. This is variable, so the LU name attribute should always be set if you are using LU 6.2.	LUNAME

Attribute	Meaning	MQSC parameter
LU6.2 member name suffix	Type the suffix of the APPCPM member of SYS1.PARMLIB. This suffix nominates the LUADD for this channel initiator.	
TCP name	Type the name of either the only, or the default, TCP/IP system to be used, depending on the value of the TCP stack type attribute. This is the name of the z/OS UNIX System Services stack for TCP/IP, as specified in the SUBFILESYSTYPE NAME parameter in the BPXPRMxx member of SYS1.PARMLIB. The default value is TCPIP. The maximum length of this parameter is eight characters. Changes to this parameter take effect when the channel initiator is restarted.	TCPNAME
Channel initiator service parm	This parameter is reserved for use by IBM.	CHISERV

TCP (Multiplatforms)



The following table lists the attributes that you can set on the **TCP** page of the Queue Manager properties dialog. If the queue manager uses the TCP/IP transport protocol to communicate with other queue managers, edit the attributes on the **TCP** page. The attributes on the **TCP** page relate to stanzas in the configuration files.

Attribute	Meaning	Stanza key
*TCP port	Type the port number for TCP/IP sessions. The default is 1414. The TCP port setting sets the qm.ini TCP port stanza for the queue manager. This is used to control the two following things: 1. A channel with a CONNAME that does not specify a port number uses this port number. 2. The command: <code>runmqclsr -t tcp -m YOUR_QM_NAME</code> uses this port number where <code>YOUR_QM_NAME</code> is the name of your queue manager.	Port
*TCP library 1	Type the name of the TCP/IP socket's DLL. The default is WSOCK32.	Library1

Attribute	Meaning	Stanza key
*TCP library 2	If there are two TCP/IP sockets, type the name of the second TCP/IP socket's DLL; if there is only one TCP/IP socket, type the same name as for the TCP library 1 attribute. The default is WSOCK32.	Library2
*TCP keepalive	TCP can check periodically that the other end of the connection is still available. If the connection is not still available, the connection is closed. To configure TCP to perform these checks, click YES ; to prevent TCP performing these checks, click NO . The default is YES.	KeepAlive
*TCP listener backlog	Type the maximum number of outstanding connection requests. The default value is -1 which resolves to the default value on the operating system. The default is 100 on Windows and Linux (x86 and x86-64 platforms).	ListenerBackLog

LU6.2 (Multiplatforms)



The following table lists the attributes that you can set on the **LU6.2** page of the Queue Manager properties dialog. If the queue manager uses the LU 6.2 transport protocol to communicate with other queue managers, edit the attributes on the **LU6.2** page. The attributes on the **LU6.2** page relate to stanzas in the configuration files.

Attribute	Meaning	Stanza key
*LU6.2 TP name	Type the TP name to start on the remote site.	TPName
*LU6.2 library 1	Type the name of the APPC DLL. The default is WCPIC32.	Library1
*LU6.2 library 2	If there are two APPCs, type the name of the second APPC DLL; if there is only one APPC, type the same name as for the LU6.2 library 1 attribute. The default is WCPIC32.	Library2
*LU6.2 local LU	Type the name of the logical unit to use on local systems.	LocalLU

NetBIOS (Multiplatforms)



The following table lists the attributes that you can set on the **NetBIOS** page of the Queue Manager properties dialog. If the queue manager uses the NetBIOS transport protocol to communicate with other queue managers, edit the attributes on the **NetBIOS** page. The attributes on the **NetBIOS** page relate to stanzas in the configuration files.

Attribute	Meaning	Stanza key
*NetBIOS local name	Type the name by which this computer will be known on the local area network (LAN).	LocalName
*NetBIOS number of sessions	Type the number of sessions to allocate. The default is 1.	NumSession
*NetBIOS number of names	Type the number of names to allocate. The default is 1.	NumNames
*NetBIOS adapter number	Type the number of the LAN adapter to use. The default is 0.	AdapterNum
*NetBIOS number of commands	Type the number of commands to allocate. The default is 1.	NumCommands
*NetBIOS library 1	Type the name of the NetBIOS DLL. The default is NETAPI32.	Library1

SPX (Multiplatforms)



The following table lists the attributes that you can set on the **SPX** page of the Queue Manager properties dialog. If the queue manager uses the SPX transport protocol to communicate with other queue managers, edit the attributes on the **SPX** page. The attributes on the **SPX** page relate to stanzas in the configuration files.

Attribute	Meaning	Stanza key
*SPX socket	Type the SPX socket number in hex. The default is SE86.	Socket
*SPX library 1	Type the name of the SPX DLL. The default is WSOCK32.	Library1
*SPX library 2	If there is a second SPX, type the name of the second SPX DLL; if there is only one SPX, type the same name as for the SPX library 1 attribute. The default is WSOCK32.	Library2
*SPX keepalive	SPX can check periodically that the other end of the connection is still available. If the connection is not still available, the connection is closed. To configure SPX to perform these checks, click YES ; to prevent SPX performing these checks, click NO . The default is YES.	KeepAlive
*SPX board number	Type the number of the LAN adapter to use. The default is 0.	BoardNum

Publish/Subscribe

The following table lists the attributes that you can set on the **Publish/Subscribe** page of the Queue Manager properties dialog. The **Publish/Subscribe** page replaces the **cfgmqbrk** application that was supplied with previous versions of IBM MQ. To configure the queue manager for publish/subscribe

messaging, edit the attributes on the **Publish/Subscribe** page. The attributes on the **Publish/Subscribe** page relate to stanzas in the configuration files. For more information about the individual stanzas, see [Configuring services and components in the IBM MQ online product documentation](#).

Attribute	Meaning	MQSC Parameter
Publish/Subscribe mode	<p>The Publish/Subscribe mode is used to allow coexistence with Publish/Subscribe engines in earlier versions of the product. The three options are:</p> <p>Compatibility which means that the V7 Publish/Subscribe engine is enabled, but the queued Publish/Subscribe interface is disabled. This means that the V7 Publish/Subscribe engine can coexist with the existing Publish/Subscribe engines. This is the default value for existing queue managers.</p> <p>Enable which means that the V7 Publish/Subscribe engine is enabled and the queued Publish/Subscribe interface is enabled. This is the default value for newly created queue managers.</p> <p>Disabled which means that all Publish/Subscribe functions are disabled.</p>	PSMODE
Message Retry Count	<p>The number of times that the channel retries to connect to the remote queue manager before it decides that it cannot deliver the message to the remote queue. This attribute controls the action of the MCA only if the Message retry exit name attribute is blank. If the Message retry exit name attribute is not blank, the value of the Message retry count attribute is passed to the exit for the exit's use but the number of times that the channel retries to connect is controlled by the exit, not by the Message retry count attribute. The maximum value is 999999999, and the default value is 5.</p>	MRRTY

Attribute	Meaning	MQSC Parameter
Publish/Subscribe syncpoint	<p>This option defines whether messages will be processed under syncpoint. The two options are:</p> <p>If persistent. The message is processed under syncpoint if the message is persistent. This is the default value.</p> <p>Yes. All messages are processed under syncpoint.</p>	PSSYNCPT
Undelivered non-persistent input message	<p>This property defines what the Pub/Sub engine should do with non-persistent input messages that are not delivered. The two options are:</p> <p>Discard. The undelivered non-persistent message is discarded. This is the default value.</p> <p>Keep. The undelivered non-persistent message is not discarded. The Pub/Sub Engine will continue to retry to process this message at appropriate intervals and does not continue processing subsequent messages.</p>	PSNPMSG

Attribute	Meaning	MQSC Parameter
Undelivered non-persistent response	<p>This property defines what the Pub/Sub engine should do with non-persistent responses that are not delivered. The four options are:</p> <p>Discard. The undelivered non-persistent response is discarded if it cannot be placed on the reply queue.</p> <p>Keep. The undelivered non-persistent response is not discarded or put on the dead-letter queue. The Pub/Sub Engine will back out the current operation and then retry it at appropriate intervals and does not continue processing subsequent messages.</p> <p>Normal. Non-persistent responses which cannot be placed on the reply queue are put on the dead-letter queue. If they cannot be placed on the DLQ then they are discarded. This is the default value.</p> <p>Safe. Non-persistent responses which 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 then the Pub/Sub Engine will back out of the current operation and then retry at appropriate intervals and does not continue processing subsequent messages.</p>	PSNPRES
Tree lifetime	<p>The lifetime, in seconds of non-administrative topics. When this non-administrative node no longer has any active subscriptions, this parameter determines how long the queue manager will wait before removing that node.</p> <p>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 through 604000. A value of 0 means that non-administrative topics are not removed by the queue manager. The queue manager's initial default value is 1800.</p>	TREELIFE

Attribute	Meaning	MQSC Parameter
Parent	<p>The name of the parent queue manager to which the local queue manager is to connect as its child in a hierarchy. If the field is left empty, then the queue manager has no parent queue manager, and if there is an existing parent queue manager it is disconnected.</p> <p>Before a queue manager can connect to a queue manager as its child in a hierarchy, channels must exist in both directions, between the parent queue manager and the child queue manager.</p>	PARENT
Publish exit path	The module name containing the publish exit code. The maximum length of this field is 128 characters. The default is no publish exit.	N/A
Publish exit function	The name of the function entry point into the module containing the publish exit code. The maximum length of this field is 128 characters.	N/A
Publish exit data	If the queue manager is using a publish exit, it invokes the exit passing an MQPSXP structure as input. The data specified using this attribute is provided in the ExitData field. The maximum length of this field is 128 characters. The default is 32 blank characters.	N/A
Publish/Subscribe clustering	<p>Controls whether this queue manager participates in Publish/Subscribe clustering. The two options are:</p> <p>Enabled which means that this queue manager can participate in Publish/Subscribe clustering. This is the default value for newly created queue managers.</p> <p>Disabled which means that this queue manager cannot participate in Publish/Subscribe clustering.</p>	PSCLUS

Archive (z/OS)



The following table lists the queue manager's system log archive attributes, or parameters, which are displayed in the **Initial** table of the queue manager's Archive dialog. The values in the **Initial** table were applied when the queue manager loaded the system parameter module at startup. You can temporarily

change and override some of the values while the queue manager is running; the new values are displayed in the **Set** table. The parameters that you can override are marked with an asterisk (*). For details of the attributes in the **Archive tape record** table, see [Archive tape](#).

The equivalent MQSC attribute for the SET ARCHIVE command is shown for each parameter. For more information about the SET ARCHIVE command, see [SET ARCHIVE](#) in the IBM MQ online product documentation.

Attribute	Meaning	MQSC parameter
Parameter type	This attribute shows which type of information is displayed in the table. The Initial table displays the initial values that were applied when the queue manager loaded the system parameter module at startup. The Set table displays any values that have been manually overridden since the queue manager started.	(Not applicable.)
*Allocation unit	Specify the unit in which primary and secondary space allocations are made: Cylinders , Tracks , or Blocks . The default is Blocks .	ALCUNIT
*Archive prefix 1	Specify the prefix for the first archive log data set name. See the <code>Time stamp format</code> attribute for a description of how the data sets are named and for restrictions on the length of the prefix. The default prefix is CSQARC1.	ARCPFX1
*Archive prefix 2	Specify the prefix for the second archive log data set name. See the <code>Time stamp format</code> attribute for a description of how the data sets are named and for restrictions on the length of the prefix. The default prefix is CSQARC2.	ARCPFX2
*Archive retention period	Specify the retention period, in days from 0 to 9999, to be used when the archive log data set is created. The default is 9999.	ARCRETN
*Routing code	Specify the list of z/OS routing codes for messages to the operator about archive log data sets. This field is ignored if the value of the <code>Waiting for reply</code> attribute is No. Type up to 14 routing codes, each with a value from 1 to 16, separated by commas. Specify at least one code.	ARCWRTC

Attribute	Meaning	MQSC parameter
*Waiting for reply	Specify whether a message is to be sent to the operator and a reply is received before attempting to mount an archive log data set. Other IBM MQ users might be forced to wait until the data set is mounted, but they are not affected while IBM MQ is waiting for the reply to the message. If the device needs a long time to mount archive log data sets, for example a tape drive, click Yes ; If the device does not have long delays, for example DASD, click No .	ARCWTOR
*Block size	Type the block size, from 4097 to 28672 (rounded up to the nearest multiple of 4 096), of the archive log data set. The block size must be compatible with the device type that you specify for the Archive unit 1 attribute. This parameter is ignored for data sets that are managed by the storage management subsystem (SMS).	BLKSIZE
*Catalog	Specify whether archive log data sets are cataloged in the primary integrated catalog facility (ICF) catalog. To archive log data sets that are cataloged, click Yes ; to archive log data sets that are not cataloged, click No . The default is No .	CATALOG
*Compact	Specify whether data written to archive logs is to be compacted. This option applies only to a 3480 or 3490 device that has the improved data recording capability (IDRC) feature. When this feature is turned on, hardware in the tape control unit writes data at a much higher density than normal, allowing for more data on each volume. If you want the data to be compacted, click Yes ; if you do not use a 3480 device with the IDRC feature or a 3490 base model, with the exception of the 3490E, click No .	COMPACT

Attribute	Meaning	MQSC parameter
*Primary space allocation	Specify the primary space allocation for DASD data sets in the unit that you specified in the <code>Allocation units</code> attribute. The value must be from 1 to 999. See <i>z/OS System Setup Guide</i> for a guide to determining the necessary value.	PRIQTY
*Secondary space allocation	Specify the secondary space allocation for DASD data sets in the unit that you specified in the <code>Allocation units</code> attribute. The value must be greater than zero.	SECQTY
*Protect	Specify whether archive log data sets are to be protected by discrete ESM (external security manager) profiles when the data sets are created. Click Yes to create discrete data set profiles after the logs offload process ends. ESM protection must be active for IBM MQ, the user ID associated with the IBM MQ queue manager address space must have authority to create these profiles, and the TAPEVOL class must be active if you are archiving to tape. If you do not want to create profiles, click No .	PROTECT
*Quiesce interval	Specify the maximum number of seconds, from 1 to 999, allowed for the quiesce when an ARCHIVE LOG command is issued with <code>MODE(QUIESCE)</code> specified.	QUIESCE
*Time stamp format	Specify whether the archive log data set name has a time stamp in it. To include a time stamp in the name, click Yes or Extended , depending on the format that you want to use. If you do not want to include a time stamp in the name, click No . For more information about the formats, see SET ARCHIVE in the IBM MQ online product documentation.	TSTAMP
*Archive unit 1	Specify the device type or unit name of the device that is used to store the first copy of the archive log data set.	UNIT
*Archive unit 2	Specify the device type or unit name of the device that is used to store the second copy of the archive log data set.	UNIT2

Archive tape (z/OS)



The following table lists the archive tape attributes, which are used in the queue manager's archive tape records. The archive tape records are listed in the **Archive tape records** table in the queue manager's Archive dialog. You cannot edit these values.

Parameter	Meaning
Parameter type	This attribute shows which type of information is displayed in the table. The Initial table displays the initial values that were applied when the queue manager loaded the system parameter module at startup. The Set table displays any values that have been manually overridden since the queue manager started.
Tape unit address	The physical address of the tape unit that is allocated to read the archive log.
Tape unit status	The status of the tape unit. Busy means that the tape unit is busy actively processing an archive log data set; Premount means that the tape unit is active and allocated for premounting; Available means that the tape unit is available, inactive, and waiting for work; Unknown means that the status of the tape unit is unknown.
Log correlation ID	The correlation ID associated with the user of the tape that is being processed. This is blank if there is no current user.
Tape volume serial number	The volume serial number of the tape that is mounted.
Data set name	The data set name on the tape volume that is being processed, or was last processed.

Statistics monitoring (z/OS)



The following table lists the attributes that you can set on the **Statistics monitoring** page of the Queue Manager properties dialog. To collect statistical data on the activity of the queue manager, edit the attributes on the **Statistics monitoring** page.

Attribute	Meaning	MQSC parameter
Channel statistics	This attribute specifies whether to collect statistics data about the activity of channels that are hosted by the queue manager. The statistics data is written to SMF. To switch off statistics data collection for the channels of the queue manager that have the value Queue Manager in their Channel statistics attribute, click Off . To switch off statistics data collection for all the queue manager's channels regardless of the setting of the channel's Channel statistics attribute, click None . To enable data collection for channels that have the value Queue Manager in their Channel statistics attribute, plus data collection for server-connection channels, click Low , Medium , or High . For more information about editing the Statistics page channel attributes, see “Statistics page” on page 454.	STATCHL
Auto CLUSSDR Statistics	This attribute specifies whether to collect statistics data about the activity of auto-defined cluster-sender channels. The statistics data is written to SMF. To inherit from the value of the queue manager's Channel statistics attribute, click Queue Manager ; to switch off data collection for auto-defined cluster-sender channels on the queue manager, click None ; to enable data collection, click Low , Medium ; or High .	STATACLS

Accounting monitoring (z/OS)



The following table lists the attributes that you can set on the **Accounting monitoring** page of the Queue Manager properties dialog. To collect data about the activity of a connection, edit the attributes on the **Accounting monitoring** page.

Attribute	Meaning	MQSC parameter
Queue accounting	This attribute specifies whether to collect accounting data about the activity of connections for queues hosted by the queue manager. To enable accounting data collection for the queue manager's queues, click On ; to disable accounting data collection for the queue manager's queues that have the value Queue Manager in their Queue accounting attribute (see Queue properties), click Off ; to disable accounting data collection for all the queue manager's queues regardless of the setting of the queue's Queue accounting attribute, click None .	ACCTQ

Log (z/OS)



The following table lists the queue manager's system log attributes, or parameters, which are displayed in the **Initial** table of the queue manager's Log dialog. The values in the **Initial** table were applied when the queue manager loaded the system parameter module at startup. You can temporarily change and override some of the values while the queue manager is running; the new values are displayed in the **Set** table. The parameters that you can override are marked with an asterisk (*). For details of the attributes in the **Log copy record** table, see [Log copy](#).

The equivalent MQSC attribute for the SET LOG command is shown for each parameter. For more information about the SET LOG command, see [SET LOG](#) in the IBM MQ online product documentation.

Parameter	Meaning	MQSC parameter
Parameter type	This attribute shows which type of information is displayed in the table. The Initial table displays the initial values that were applied when the queue manager loaded the system parameter module at startup. The Set table displays any values that have been manually overridden since the queue manager started.	(Not applicable.)
*Deallocation interval	Specify the length of time, in minutes, that an allocated archive read tape unit is allowed to remain unused before it is deallocated. The value can be from zero to 1440. If the value is zero, the tape unit is deallocated immediately; if the value is 1440, the tape unit is never deallocated.	DEALLCT

Parameter	Meaning	MQSC parameter
*Log compression	<p>Specifies the log data compression technique for persistent message logging.</p> <p>NONE means that log compression is not enabled. This is the default value.</p> <p>RLE means that run-length encoding log compression is enabled.</p> <p>ANY means that any compression algorithm supported by the queue manager is enabled. In IBM WebSphere MQ 7.0.1 and later, the only compression algorithm available for value ANY is RLE.</p>	COMPLOG
*Maximum number of log archives	Specify the maximum number of archive log volumes that can be recorded in the BSDS.	MAXARCH
*Maximum number of tape units	Specify the maximum number of dedicated tape units that can be allocated to read archive log tape volumes. This overrides the value for MAXRTU set by CSQ6LOGP in the archive system parameters. This, together with the Deallocation interval attribute, allows IBM MQ to optimize archive log reading from tape devices.	MAXRTU
Input buffer size	Specifies the size of input buffer storage for active and archive log data sets.	INBUFF
Output buffer size	Specifies the size of output buffer storage for active and archive log data sets.	OUTBUFF
*Output buffer count	Specifies the number of output buffers to be filled before they are written to the active log data sets.	WRTHRSH
Log archive	Specifies whether archiving is on or off. Yes means that archiving is on; No means that archiving is off.	OFFLOAD
Dual logging used	Specifies whether dual logging is being used. Yes means that dual logging is being used; No means that dual logging is not being used.	TWOACTV

Parameter	Meaning	MQSC parameter
Dual archive logging used	Specifies whether dual archive logging is being used. Yes means that dual archive logging is being used; No means that dual archive logging is not being used.	TWOARCH
Dual BSDS used	Specifies whether dual BSDS is being used. Yes means that dual BSDS is being used; No means that dual BSDS is not being used.	TWOBSDS

Log copy (z/OS)




The following table lists the log copy attributes, which are used in the queue manager's log copy records. The log copy records are listed in the **Log copy records** table in the queue manager's Log dialog. You cannot edit these values.

Parameter	Meaning
Log copy number	The number of the copy.
Log used	The percentage of the active log data set that has been used.
Data set name	The data set name of the active log data set. If the copy is not currently active, this is returned as blank.

Security (z/OS)



The following table lists the queue manager's system-wide security attributes, or parameters. You can change two of the values; the parameters that you can change are marked with an asterisk (*). For details of the attributes in the **Security switch** table, see [Security switch](#).

The equivalent MQSC attribute for the ALTER SECURITY command is shown for each parameter. For more information about the ALTER SECURITY command, see  [ALTER SECURITY](#) in the IBM MQ online product documentation.

Parameter	Meaning	MQSC parameter
*Security timeout	Type, in minutes from 0 to 10080 (one week), how long security information about an unused user ID and associated resources is retained by IBM MQ. If you type 0, and the value of the Security interval attribute is not zero, all such information is discarded by the queue manager every Security interval.	TIMEOUT

Parameter	Meaning	MQSC parameter
*Security interval	Type, in minutes from 0 to 10080 (one week), the interval between checks for user IDs and their associated resources to determine whether the Security timeout has expired. If you type 0, no user timeouts occur.	INTERVAL

Security switch (z/OS)



The following table lists the security switch attributes, which are used in the queue manager's security switch messages. The security switch messages (one per security switch) are listed in the **Security switch** table in the queue manager's Security dialog. You cannot edit these values.

Parameter	Meaning
Security switch	The name of the security switch.
Security setting	The current setting of the security switch, and whether the profile that caused the setting is present. For example, the security switch could be set off because the relevant profile was not found.
Security profile	The name of the profile that caused the current security setting.


System (z/OS)



The following table lists the queue manager's system attributes, or parameters, which are displayed in the **Initial** table of the queue manager's System dialog. The values in the **Initial** table were applied when the queue manager loaded the system parameter module at startup. You can temporarily change and override some of the values while the queue manager is running; the new values are displayed in the **Set** table. The parameters that you can override are marked with an asterisk (*).

The equivalent MQSC attribute for the DISPLAY SYSTEM command is shown for each parameter. For more information about the DISPLAY SYSTEM command, see [DISPLAY SYSTEM](#) in the IBM MQ online product documentation.

Parameter	Meaning	MQSC parameter
Parameter type	This attribute shows which type of information is displayed in the table. The Initial table displays the initial values that were applied when the queue manager loaded the system parameter module at startup. The Set table displays any values that have been manually overridden since the queue manager started.	(Not applicable.)

Parameter	Meaning	MQSC parameter
Connection swap	Whether or not batch jobs can currently be swapped out during some MQ API calls.  Attention: From IBM MQ 9.0, this keyword has no effect.	CONNSWAP
Command user ID	Specifies the default user ID for command security checks.	CMDUSER
*Excluded operator messages	A list of messages excluded from being written to any log.	EXCLMSG
Exit interval	Specifies the time, in seconds, for which queue manager exits can execute during each invocation.	EXITLIM
Exit tasks	Specifies how many started server tasks to use to run queue manager exits.	EXITTCB
*Checkpoint count	Specify the number of log records, from 200 to 16000000, that IBM MQ writes between the start of one checkpoint and the next. IBM MQ starts a new checkpoint after the number of records that you specify has been written.	LOGLOAD
XCF group name	Specifies the name of the XCF group to which this instance of IBM MQ belongs.	OTMACON = (Group)
XCF member name	Specifies the name of the XCF member to which this instance of IBM MQ belongs.	OTMACON = (Member)
OTMA exit name	Specifies the name of the OTMA destination resolution user exit to be run by IMS.	OTMACON = (Druexit)
OTMA interval	Specifies the length of time, in seconds, that a user ID from IBM MQ is considered previously verified by IMS.	OTMACON = (Age)
OTMA Tpipe name prefix	Specifies the prefix to be used for Tpipe names.	OTMACON = (Tpipepfx)

Parameter	Meaning	MQSC parameter
Defer index	Specifies whether the queue manager restart completes before all indexes are built and defers building until later, or whether the queue manager waits until all indexes are built. Yes means that the queue manager restart completes before all indexes are built; No means that the queue manager restart waits until all indexes are built.	QINDXBLD
Coded character set ID	Specifies the coded character set identifier for the queue manager.	QMCCSID
Queue sharing group name	Specifies the name of the queue sharing group to which the queue manager belongs.	(Not applicable.)
Data-sharing group name	Specifies the name of the Db2 data-sharing group to which the queue manager is to connect.	(Not applicable.)
Db2 name	Specifies the name of the Db2 subsystem or group attachment to which the queue manager is to connect.	(Not applicable.)
Db2 tasks	Specifies the number of Db2 server tasks to use.	(Not applicable.)
Db2 BLOB tasks	Specifies the number of Db2 server tasks to be used for BLOBs.	(Not applicable.)
Write RACF audit records	Specifies whether RACF audit records are written for RESLEVEL security checks that are performed during connection processing. Yes means that RACF audit records are written; No means that RACF audit records are not written.	RESAUDIT
Routing code	Specifies the list of z/OS routing codes for messages that are not sent in direct response to an MQSC command. There can be 1 - 16 entries in the list.	ROUTCDE
Send accounting data to SMF	Specifies whether IBM MQ sends accounting data to SMF automatically when the queue manager starts. Yes means that accounting data is sent automatically; No means that accounting data is not sent automatically.	SMFACCT

Parameter	Meaning	MQSC parameter
Send statistics data to SMF	Specifies whether IBM MQ sends statistics data to SMF automatically when the queue manager starts. Yes means that statistics data is sent automatically; No means that statistics data is not sent automatically.	SMFSTAT
*SMF interval	Specify the interval, in minutes from 0 to 1440, between consecutive gatherings of statistics. If you specify a value of zero, both statistics data and accounting data is collected at the SMF data collection broadcast. Changes to this parameter take effect when the current interval expires, unless the new interval is less than the unexpired portion of the current interval, in which case statistics are gathered immediately and the new interval then takes effect.	STATIME
Trace classes	Specifies the classes for which tracing is started automatically.	TRACSTR
*Trace table size	Specify the default size, in 4 KB blocks from 1 to 999, of trace table where the global trace facility stores IBM MQ trace records. Storage for the trace table is allocated in the ECSA so you must select this value with care. If there is any trace currently in effect, the existing trace table continues to be used, and its size is unchanged. A new global trace table is only obtained when trace is restarted. If a new trace table is created with insufficient storage, the old trace table continues to be used, and the message CSQW153E is displayed.	TRACTBL
Cluster cache type	Specifies the type of cluster cache. Static means that a static cluster cache is used; Dynamic means that a dynamic cluster cache is used.	(Not applicable.)
WLM interval	Specifies the time, in minutes, between scans of the queue index for WLM-managed queues.	WLMTIME
WLM units	Indicates whether WLM interval is in seconds or minutes.	WLMTIMU
*Service parameter setting	This parameter is reserved for use by IBM.	SERVICE

Parameter	Meaning	MQSC parameter
Operation Mode	Specifies the operation mode for the queue manager. This attribute can have a value of COMPAT or NEWFUNC, followed by a command level value in the format <i>vrn</i> . For example, 600, 700, or 701. The attribute is read only and can not be changed by the GUI. When this value is set to COMPAT, the user will not be able to access certain function associated with the current installed level and will result in a MQRCCF_FUNCTION_RESTRICTED error.	OPMODE = (COMPAT,701)
Security policies	Indicates whether the security capabilities of Advanced Message Security are available.	SPLCAP
Maximum ACE pool size (KB)	The maximum size of the ACE storage pool in KB in the range 0 - 999 999. An ACE is required for each connected application, and some types of application require extra ACEs for processing. Internal queue manager threads also require them. The ACE storage pool is allocated in ECSA. For queue managers that use a large quantity of ECSA storage, the ECSA storage allocation grows linearly with the size of the ACE storage pool. A value of zero for this parameter means that there is no limit on the size of the ACE storage pool. In extreme circumstances, the ACE storage pool can use all of the available ECSA storage, resulting in a system outage for the LPAR.	ACELIM

Related tasks

[“Configuring queue managers and objects” on page 51](#)

You can configure many of the attributes of queue managers and their objects from IBM MQ Explorer using the properties dialogs.

Related reference

[“Strings in property dialogs” on page 655](#)

To include certain characters in a string, you must punctuate the string in a particular way.

IBM MQ queue properties

The attributes that you can set for a queue depend on the type of queue. Different types of IBM MQ queues have different properties. Some of the attributes do not apply to all types of queue, some attributes are specific to cluster queues, and some attributes are specific to z/OS queues.

The following tables list the attributes that you can set for all types of queues:

- [General](#)
- [Extended](#)
- [Cluster](#)
- [Triggering](#)
- [Events](#)
- [Storage](#)
- [Statistics](#)

For each attribute, there is a brief description of when you might need to configure the attribute. The tables also give the equivalent MQSC parameter for the ALTER QUEUE and DISPLAY QUEUE commands. For more information about MQSC commands, see [The MQSC Commands](#) in the online IBM MQ product documentation.

General page

The following table lists the attributes you can set on the **General** page of the **Queue properties** dialog.

Attribute	Description	MQSC parameter
Queue name	Read-only. You cannot change the name of the queue after it has been created.	QNAME
Queue type	Read-only. You cannot change the type of the queue after it has been created.	QTYPE
QSG disposition	(z/OS shared queue only) Read-only. The queue sharing group disposition of the queue. Specifies the disposition of the object (where it is defined and how it behaves). You cannot change the disposition of a queue after it has been created. Queue manager means that the object definition is available only to the queue manager that hosts it; Group means that the object definition is stored on the shared repository and each queue manager in the queue sharing group has a copy of the definition; Copy means that the object definition is the queue manager's copy of a definition in the shared repository; Shared means that the object definition is stored on the queue-sharing group's coupling facility and is available to all the queue managers in the queue sharing group.	QSGDISP
Description	Type a meaningful description of the purpose of the queue. See “Strings in property dialogs” on page 655 .	DESCR

Attribute	Description	MQSC parameter
Put messages	To enable messages to be put on the queue, select Allowed ; to prevent messages from being put on the queue, select Inhibited .	PUT
Get messages	To enable messages to be got from the queue, select Allowed ; to prevent messages being got from the queue, select Inhibited .	GET
Default priority	Type the default priority, from 0 - 9, of messages that are put on the queue; 0 is the lowest priority.	DEFPRTY
Default persistence	The default persistence of a new queue is Not persistent. Select Persistent to specify that messages created by applications that use MQPER_PERSISTENCE_AS_Q_DEF become persistent. Select Not Persistent to specify that messages created by applications that use MQPER_PERSISTENCE_AS_Q_DEF become non persistent.	DEFPSIST
Scope	To put the queue in the cell directory and make the queue known to all the queue managers within the cell, select Cell ; to restrict the queue's scope so it does not extend beyond its queue manager, select Queue Manager .	SCOPE
Usage	To make the queue a local queue, select Normal ; to make the queue a transmission queue, select Transmission . Do not change the Usage attribute while there are messages on the queue.	USAGE
Base type	Select the type of object (Queue or Topic) to which the alias queue resolves. The default value is Queue .	TARGETTYPE
Remote queue	Type the name of the queue to which the remote queue definition points.	RNAME
Remote Queue manager	Type the name of the queue manager that hosts the remote queue.	RQMNAME
Transmission queue	Type the name of the transmission queue that the local queue manager uses to send messages to the remote queue manager.	XMITQ

Extended page

The following table lists the attributes you can set on the **Extended** page of the **Queue properties** dialog.

Attribute	Description	MQSC parameter
Max queue depth	Type the maximum number of messages that are allowed on the queue. Specify a value from 0 - 999999999.	MAXDEPTH
Max message length	Type the maximum length of a message, in bytes, that is allowed on the queue. On all platforms other than z/OS, specify a value from 0 to the maximum message length for the queue manager. See the Maximum message length attribute in Queue manager properties . On z/OS, specify a value from 0 - 100 MB. If the value of the QSG disposition attribute is Shared, or if the value of the Definition type attribute is Shared dynamic, then the Max message length value for the queue must be from 0 - 4 MB. On other platforms, specify a value from 0 - 4 MB.	MAXMSGL
Shareability	To share the queue so that more than one instance of an application can open this queue for input, select Shareable ; to restrict the queue so that only one instance of an application can open the queue at a time, select Not Shareable .	SHARE
Default input open option	To allow applications that open the queue for input to have exclusive access to the messages on the queue, select Exclusive ; to allow any number of applications that open the queue for input to access the messages on the queue, select Shared .	DEFSOPT
Message delivery sequence	To specify to get messages from the queue in order of their priority, select Priority . To specify to get messages from the queue in the order that they were put on the queue, select FIFO (first in, first out).	MSGDLVSQ
Retention interval	Type the number of hours from the date and time when it was created (from 0 - 999999999) that the queue might be needed. You can use this information to determine when the queue is no longer required. The queue will not be deleted when it is no longer required.	RETINTVL
Pipe name	(z/OS shared queue only) Read only.	
Index type	(z/OS shared queue only) To specify the type of index that the queue manager maintains to increase the speed of MQGET operations on the queue, select one of these five options: None: No index is maintained. Use this when retrieving messages sequentially. This is the default. Group ID: An index of group identifiers is maintained. You must use this index type if you want logical ordering of message groups. Correl ID: An index of correlation identifiers is maintained. Use this when retrieving messages using the CorrelId field as a selection criterion on the MQGET call. Message ID: An index of message identifiers is maintained. Use this when retrieving messages using the MsgId field as a selection criterion on the MQGET call. Message token: An index of message tokens is maintained.	

Attribute	Description	MQSC parameter
Definition type	<p>For local queues, this attribute is read-only: Predefined means that the queue was created by an operator or an authorized application sending a command message to the service queue; Permanent dynamic means that the queue was created by an application issuing an MQOPEN call with the name of a model queue specified in the object descriptor (MQOD) and the queue is permanent; Temporary dynamic means that the queue was created by an application issuing an MQOPEN call, but the queue is temporary; Shared dynamic (z/OS only) also means that the queue was created by an application issuing an MQOPEN call, but the queue is permanent and has the queue sharing group disposition of Shared.</p> <p>For model queues, this attribute is editable; To specify that a permanent dynamic queue is created from this model queue, select Permanent dynamic (on z/OS, the dynamic queue has a disposition of Queue manager); to specify that a temporary dynamic queue is created, select Temporary dynamic (on z/OS, the dynamic queue has a disposition of Queue manager); on z/OS only, to specify that a permanent dynamic queue is created with a disposition of Shared, select Shared dynamic.</p>	DEFTYPE
Default read ahead	<p>To configure read ahead at the queue level, select Yes. Non-persistent messages are automatically read ahead by the client, ahead of the application requesting them. Non-persistent messages might be lost if the client ends abnormally or if the client application does not consume all the messages it is sent.</p> <p>To configure the queue so that non-persistent messages are not automatically read ahead by the client, select No. This is the default value. Messages are not automatically read ahead by the client ahead of an application requesting them. Messages are only read ahead if requested by the client. A maximum of one non-persistent message might be lost if the client ends abnormally.</p> <p>To disable read ahead at the queue level, select Disabled. Messages are not read ahead by the client ahead of an application requesting them, regardless of whether read ahead is requested by the client application.</p>	DEFREADA
Default put response type	The default response type for message puts. To specify that the response is put synchronously, select Synchronous . To specify that the response is put asynchronously, select Asynchronous .	DEFPRESP
Distribution lists	To allow distribution list messages to be put on the queue, select Enabled . To prevent distribution list messages being put on the queue, select Disabled .	DISTL

Attribute	Description	MQSC parameter
Property control (Only on Local queues, Alias queues, and Model queues)	<p>This defines what happens to properties of messages that are retrieved from queues using the MQGET command when the MQGMO_PROPERTIES_AS_Q_DEF option is specified.</p> <p>To contain all the properties of the message, except those contained in the message descriptor (or extension), select All. The All value means that all properties of the message are included with the message when it is sent to the remote queue manager. The properties, except those properties in the message descriptor (or extension), are placed in one or more MQRFH2 headers in the message data.</p> <p>To allow applications that expect JMS related properties to be in an MQRFH2 in the message data to continue to work unmodified, select Compatibility. This is the default value. Compatibility means that if the message contains a property with a prefix of mcd., jms., usr., or mqext. then all message properties are delivered to the application in an MQRFH2 header. Otherwise all properties of the message, except those contained in the message descriptor (or extension), are discarded and are no longer accessible to the application.</p> <p>To ensure that properties are always returned in the message data in an MQRFH2 header, regardless of whether the application specifies a message handle, select Force MQRFH2. A valid message handle supplied in the MsgHandle field of the MQGMO structure on the MQGET call is ignored. Properties of the message are not accessible via the message handle.</p> <p>To discard all the properties of a message, except those contained in the message descriptor (or extension), select None. This value prevents applications that do not support message properties from being affected by the inclusion of any property in a message.</p> <p>V6COMPAT - MQRFH2 headers are not modified by the properties code. If message properties have been provided and are not contained in the original MQRFH2 headers, they are returned in a message handle or otherwise discarded. This behavior can be overridden by supplying one of the MQGMO_PROPERTIES options.</p> <p>Note: For a transmission queue, which is a local queue with Usage set to Transmission, the Property Control attribute of the queue is irrelevant and it is the Property Control attribute on the corresponding channel object that controls the message property behavior.</p>	PROPCTL

Attribute	Description	MQSC parameter
Custom	<p>The Custom parameter is reserved for the configuration of new features before separate attributes have been introduced. The possible values are a list of zero or more attributes-value pairs, in MQSC-style syntax, separated by at least 1 space.</p> <p>The attribute names and values are case-sensitive, and must be specified in uppercase. The values can contain spaces and parentheses, but must not contain single quotation marks. Examples of valid syntax are:</p> <ul style="list-style-type: none"> • CUSTOM(' ') • CUSTOM('A(B)') • CUSTOM('C(D) E(F)') • CUSTOM('G(5000) H(9.20.4.6(1415))') <p>The queue manager parses the value, but if the string cannot be parsed according to these rules, or if it contains attributes or values that are not recognized, the queue manager ignores the errors.</p>	CUSTOM
Cluster channel names	<p>Set the Cluster channel names parameter on a cluster transmit queue to override the default association of cluster-sender channels with cluster transmission queues. You can specify which cluster-sender channels transfer messages from this transmission queue.</p> <p>The default is for all cluster-sender channels to transfer messages from a single cluster transmission queue, <code>SYSTEM.CLUSTER.TRANSMIT.QUEUE</code>. You can change the default for the queue manager, so that all cluster-sender channels transfer messages from separate transmission queues. The queue manager attribute is Default cluster transmission queue. The queue manager creates separate transmission queues automatically, when they are required. The queue manager does not set the Cluster channel name parameter</p> <p>Set the Cluster channel names parameter to the name of a single cluster-sender channel, or to a generic name. A generic name associates multiple cluster-sender channels with this transmission queue. A generic name has wildcard characters, "*", in any positions in the name. All cluster-sender channels that match the name transfer messages from this transmission queue and no other.</p> <p>On z/OS, if this parameter is set, the queue must be shareable, be indexed by correlation ID, and must not be a dynamic or a shared queue.</p>	CLCHNAME

Cluster page

The following table lists the attributes you can set on the **Cluster** page of the **Queue properties** dialog. To share the queue in one or more clusters, edit the attributes on the **Cluster** page.

Attribute	Description	MQSC parameter
Not shared in cluster	To specify that this queue is not available to other queue managers via cluster connections, select this option.	(Not applicable.)
Shared in cluster	To make this queue available to other queue managers in just one cluster, select this option, then type the name of the cluster.	CLUSTER
Shared in a list of clusters	To make this queue available to other queue managers in more than one cluster, select this option, then type the name of the namelist that contains the list of clusters.	CLUSNL
Default bind type	This attribute specifies the binding to be used when the application specifies MQ00_BIND_AS_Q_DEF on the MQOPEN call, and the queue is a cluster queue. To bind the queue handle to a specific instance of the cluster queue when the queue is opened, select On open ; to allow the queue manager to select a specific queue instance when the message is put using MQPUT, and to change that selection subsequently if necessary, select Not fixed .	DEFBIND
CLWL Queue rank	This attribute is the Cluster Workload (CLWL) queue rank. Type the ranking of the queue in the cluster, from 0 - 9; 0 is the lowest priority. For more information, see Distributed queuing and clusters in the IBM MQ online product documentation.	CLWLRANK
CLWL Queue priority	This attribute is the Cluster Workload (CLWL) queue priority. Type the priority of the queue in the cluster, from 0 - 9; 0 is the lowest priority. For more information, see Distributed queuing and clusters in the IBM MQ online product documentation.	CLWLPRTY

Attribute	Description	MQSC parameter
CLWL use queue	<p>The Cluster Workload (CLWL) use queue property. It defines the behavior of an MQPUT when the target queue has both a local instance and at least one remote cluster instance. If the put originates from a cluster channel, this attribute does not apply. Select one of the following options:</p> <p>Select Queue manager to use the value specified by the CLWL use queue property of the queue manager that the selected queue belongs to. This is the default value.</p> <p>Select Any to use local and remote queues.</p> <p>Select Local to use only local queues</p> <p>For more information, see Distributed queuing and clusters in the IBM MQ online product documentation.</p>	CLWLUSEQ

Triggering page

The following table lists the attributes you can set on the **Triggering** page of the **Queue properties** dialog. To configure the queue for triggering, edit the attributes on the **Triggering** page.

Attribute	Description	MQSC parameter
Trigger control	To enable triggering on the queue, select On , then configure the other trigger attributes for the queue; to disable triggering on the queue, select Off .	TRIGGER
Trigger type	To trigger an event when the current depth of the queue goes from 0 - 1, select First ; to trigger an event when the queue depth threshold is exceeded, select Depth ; to trigger an event every time a message is put on the queue, select Every .	TRIGTYPE
Trigger depth	Type the number of messages that must be put on the queue to trigger an event.	TRIGDEPTH

Attribute	Description	MQSC parameter
Trigger message priority	Type the minimum priority, from 1 - 9, that a message must have for it to count towards a trigger event. The queue manager ignores messages with a lesser priority when it determines whether to create a trigger message. To count all messages towards a trigger event, type 0.	TRIGMPRI
Trigger data	Type free-form data for the queue manager to insert into the trigger message when the trigger event is caused by this queue. The data has no significance to the queue manager. The data is meaningful to either the trigger monitor application that processes the initiation queue, or to the application that is started by the trigger monitor.	TRIGDATA
Initiation queue	Type the name of the initiation queue. When the criteria for a trigger event are met, the queue manager puts a trigger message on the initiation queue.	INITQ
Process name	<p>The local name of the IBM MQ process. This parameter is supported only on local and model queues.</p> <p>This is the name of a process instance that identifies the application started by the queue manager when a trigger event occurs. The process does not have to be defined when the local queue is defined, but it must be available for a trigger event to occur.</p> <p>If the queue is a transmission queue, then the process definition contains the name of the channel to be started. This parameter is optional, and if you do not specify a process name, then the channel name is taken from the value specified for the TRIGDATA parameter.</p>	PROCESS

Events page

The following table lists the attributes you can set on the **Events** page of the **Queue properties** dialog. To configure the queue manager to generate events in response to certain criteria on the queue, edit the attributes on the **Events** page.

Attribute	Description	MQSC parameter
Queue depth max events	To generate a Queue Full event when a message is put to the queue, but is rejected because the queue is already full, select Enabled .	QDPMAXEV
Queue depth high events	To generate a Queue Depth High event when a message is put to the queue causing the queue depth to be greater than or equal to the value of the Queue depth high limit attribute, select Enabled .	QDPHIEV
Queue depth high limit	This is the percentage value of the maximum queue depth that is used as the queue depth high limit. Type the maximum queue depth limit percentage with which the queue manager compares the current queue depth to determine whether to generate a Queue Depth High event.	QDEPTHHI
Queue depth low events	To generate a Queue Depth Low event when a message is retrieved from the queue causing the queue depth to be less than or equal to the value of the Queue depth low limit attribute, select Enabled .	QDPLOEV
Queue depth low limit	This is the percentage value of the maximum queue depth that is used as the queue depth low limit. Type the minimum queue depth limit percentage with which the queue manager compares the current queue depth to determine whether to generate a Queue Depth Low event.	QDEPTHLO
Queue service interval events	To generate a Queue Service Interval High event when a check indicates that no messages have been retrieved from the queue for at least the time indicated by the Queue service interval attribute, select High ; to generate a Queue Service Interval OK event when a check indicates that messages have been retrieved from the queue within the time indicated by the Queue service interval attribute, select OK ; to disable Queue Service Interval events, select None .	QSVCIEV

Attribute	Description	MQSC parameter
Queue service interval	Type the service interval, in milliseconds, from 0 - 999999999. The queue manager uses this value to determine whether to generate Queue Service Interval High or Queue Service Interval OK events.	QSVCINT

Storage page

The following table lists the attributes you can set on the **Storage** page of the **Queue properties** dialog. To configure how IBM MQ deals with messages that are backed out, edit the attributes on the **Storage** page.

Attribute	Description	MQSC parameter
Backout requeue queue	Type the name of the queue to which a message is transferred if it is backed out more than the number of times specified in the Backout threshold attribute.	BOQNAME
Backout threshold	Type the number of times that the message can be backed out before it is transferred to the backout queue you specified in the Backout requeue queue attribute.	BOTHRESH
Harden get backout	To ensure that the backout count (the number of times the message has been retrieved by the MQGET call and then subsequently backed out) is accurately retained if the queue manager is restarted, select Hardened . Hardening the count has a detrimental effect on performance, so select Hardened only if it is essential that the count is accurate. If it is not essential that the count is accurate, select Not hardened .	HARDENBO
NPM class	This attribute specifies the circumstances under which non-persistent messages on the queue are discarded. To specify that non-persistent messages are discarded when the queue manager is restarted, select Normal ; this is valid only for non-shared queues. To specify that the queue manager should try to persist non-persistent messages for the lifetime of the queue, select High ; this is valid for non-shared and shared queues; the non-persistent messages might still be lost in the event of a failure.	NPMCLASS

Attribute	Description	MQSC parameter
Storage class name	(z/OS shared queue only) This is the name of the storage class that maps the queue to a page set. The queue's messages are stored on that page set. You can change this attribute but the queue must first be empty and closed. The storage class name must be entered in uppercase.	STGCLASS
coupling facility name	(z/OS shared queue only) This is the name of the coupling facility structure on which the queue's messages are stored. You can change this attribute but the queue must first be empty and closed.	CFSTRUCT

Statistics page

The following table lists the attributes on the **Statistics** page of the **Queue properties** dialog. The **Statistics** page displays information about the history of the queue. Not all of the attributes can be edited.

Attribute	Description	MQSC parameter
Creation date	Read-only. This is the date when the queue was created.	CRDATE
Creation time	Read-only. This is the time at which the queue was created.	CRTIME
Open input count	Read-only. This is the number of applications that are currently connected to the queue to get messages from the queue.	IPPROCS
Open output count	Read-only. This is the number of applications that are currently connected to the queue to put messages on the queue.	OPPROCS
Current queue depth	Read-only. This is the number of messages currently on the queue.	CURDEPTH
Alteration date	Read-only. This is the date on which the queue's attributes were last altered.	ALTDATE
Alteration time	Read-only. This is the time at which the queue's attributes were last altered.	ALTTIME

Attribute	Description	MQSC parameter
Queue monitoring	<p>You can configure IBM MQ to collect online monitoring data about the current performance of the queue. To inherit the value of the queue manager's Queue monitoring attribute (see “Queue manager properties” on page 349), select Queue manager. If the queue manager's Queue monitoring attribute is None, the queue's Queue monitoring attribute is ignored.</p> <p>If the queue manager's Queue monitoring attribute is not None:</p> <ul style="list-style-type: none"> • To override the queue manager's settings and prevent data collection for this queue, select Off. • To collect data at a low rate, select Low. • To collect data at a medium rate, select Medium. • To collect data at a high rate, select High. <p>For more information see Monitoring and performance in the IBM MQ online product documentation.</p>	MONQ
Queue statistics	<p>You can configure IBM MQ to collect statistics data about the activity of the queue. To inherit the value of the queue manager's Queue statistics attribute (see “Queue manager properties” on page 349), select Queue manager. If the queue manager's Queue statistics attribute is None, the queue's Queue statistics attribute is ignored. If the queue manager's Queue statistics attribute is not None: to override the queue manager's settings and prevent data collection for this queue, select Off; to override the queue manager's settings and collect data, select On. For more information see Monitoring and performance in the IBM MQ online product documentation.</p>	STATQ

Attribute	Description	MQSC parameter
Queue accounting	You can configure IBM MQ to collect statistics data about the activity of connections for this queue. To inherit the value of the queue manager's Queue accounting attribute (see “Queue manager properties” on page 349), select Queue manager . If the queue manager's Queue accounting attribute is None, the queue's Queue accounting attribute is ignored. If the queue manager's Queue accounting attribute is not None: to override the queue manager's settings and prevent data collection for this queue, select Off ; to override the queue manager's settings and collect data, select On . For more information see Monitoring and performance in the IBM MQ online product documentation.	ACCTQ

Related concepts

[“IBM MQ queues”](#) on page 29

A queue is a container for messages. Business applications that are connected to the queue manager that hosts the queue can retrieve messages from the queue or can put messages on the queue.

Related tasks

[“Configuring queue managers and objects”](#) on page 51

You can configure many of the attributes of queue managers and their objects from IBM MQ Explorer using the properties dialogs.

[“Forcing changes to queue properties”](#) on page 53

If the changes that you are making to the attributes of a queue affect the operation of a queue manager or another program, you might be asked to confirm whether you want to force the changes to the queue attributes.

Related reference

[“Strings in property dialogs”](#) on page 655

To include certain characters in a string, you must punctuate the string in a particular way.

[“Topic properties”](#) on page 458

An IBM MQ topic is an IBM MQ object that identifies what a publication is about. You can set attributes for topics. Some topic attributes are specific to z/OS topics. Also, there are some attributes that you can alter only while you are creating a topic. You cannot modify these attributes after the IBM MQ topic has been created.

JMS Destination properties

You can view and set destination properties in the Destination properties dialog. The properties that are available in the dialog depend on the type of destination.


Channel properties

You can set properties for all types of channels, including client-connection channels. Some properties are specific to certain types of channel.

The following tables list all the properties that you can set:

- [General](#)
- [Extended](#)
- [MCA](#)
- [Exits](#)
- [LU6.2](#)
- [Retry](#)
- [Message retry](#)
- [Cluster](#)
- [SSL](#)
- [Load Balancing](#)
- [Statistics](#)

Some of the properties do not apply to all types of channel:


- Some properties are specific to cluster channels.
-  Some properties are specific to z/OS channels.

For each property, there is a brief description of when you might need to configure it. The tables also give the equivalent MQSC parameter for the DEFINE, ALTER and DISPLAY CHANNEL commands. For more information about MQSC commands, see [Administration using MQSC commands](#).

General page

The following table lists the properties that you can set on the **General** page of the **Channel properties** dialog.

Property	Meaning	MQSC parameter
Channel name	Read-only. This is the name of the channel definition.	CHANNEL
Type	Read-only. This is the type of the channel definition.	CHLTYPE
QSG disposition	Read-only. This is the queue sharing group disposition of the channel definition. You cannot change the disposition of a channel definition after it has been created. Queue manager means that the object definition is available only to the queue manager that hosts it; Group means that the object definition is stored on the shared repository and each queue manager in the queue sharing group has a copy of the definition; Copy means that the object definition is the queue manager's copy of a definition in the shared repository.	QSGDISP
Description	Type a meaningful description of the purpose of the channel. See “Strings in property dialogs” on page 655 .	DESCR


Property	Meaning	MQSC parameter
Queue manager name	Type the name of the queue manager on which the channel is defined. For client-connection channels, type the name of the queue manager to which an application that is running in the MQI client environment can request connection.	QMNAME
Transmission protocol	Select from the list the transport type that the channel uses.	TRPTYPE
Connection name	<p>For all types of channel except cluster-receiver channels, type the name of the computer that hosts the target queue manager. The format of the connection name depends on the transmission protocol that is selected. For example, if you are using the TCP/IP protocol and you know that the target queue manager is connecting using a port number other than the IBM MQ default of 1414, type <i>computer_name(port_number)</i>, where <i>computer_name</i> is the name or IP address of the computer that hosts the target queue manager, and <i>port_number</i> is the port that the target queue manager's listener is using.</p> <p> For cluster-receiver channels on Windows, UNIX and Linux, that use the TCP/IP transport protocol, do not specify a value for this property; IBM MQ generates a name for using, assuming the default port and the current IPv4 address of the system. If the system does not have an IPv4 address, the current IPv6 address of the system is used. For cluster-receiver channels on other platforms, and for cluster-receiver channels that do not use the TCP/IP transport protocol, type the name of the computer that hosts the local queue manager.</p>	CONNAME
Transmission queue	Type the name of the transmission queue that corresponds to the queue manager at the receiver end of the channel.	XMITQ

Property	Meaning	MQSC parameter
Local communication address	<p>If the channel uses TCP/IP and you want the channel to use a particular IP address, port, or port range for outbound communications, type the local communications address for the channel. The channel binds to the address locally. Use the format <i>ipaddress(low-port, high-port)</i>, where <i>ipaddress</i> is the IP address specified in IPv4 dotted decimal, IPv6 hexadecimal, or alphanumeric host name format. For example, 192.0.2.0 specifies the IPv4 address with any port; 192.0.2.0(1000) specifies the IPv4 address and a specific port; 192.0.2.0(1000,2000) specifies the IPv4 address and a range of ports; (1000) specifies a port only.</p> <p>Cluster-sender channels: If you type a value in the Local communication address field of a manually defined cluster-sender channel, this value is overwritten with the values in the full repository's cluster-receiver channel when communication is established with the full repository queue manager. As well as specifying the value in the manually defined cluster-sender channel, you must write a channel auto-definition exit to force the value of the Local communication address property into any automatically defined cluster-sender channels.</p> <p>Cluster-receiver channels: Do not put an IP address in the Local communication address field of a cluster-receiver channel unless all of the queue managers are on the same computer. This is because any queue manager that attempts to connect to a queue manager with an IP address in the Local communication address field of its cluster-receiver channel has these values propagated to their auto-defined cluster-sender channels. You can, however, put a port number or port range in the Local communication address field of a cluster-receiver channel if you want all the queue managers in a cluster to use a specific port or range of ports for all</p>	<p>LOCLADDR</p> <p>Introduction to IBM MQ Explorer 429</p>


Property	Meaning	MQSC parameter
Overall channel status	Read-only. This is the status of the channel.	STATUS


Extended page

The following table lists the properties that you can set on the **Extended** page of the **Channel Properties** dialog.

Property	Meaning	MQSC parameter
Maximum message length	<p>Type the maximum length of a message that can be transmitted on the channel:</p> <ul style="list-style-type: none"> On AIX, HP-UX, IBM i, Solaris, Windows, and VSE/ESA, the value must be greater than or equal to zero, and less than or equal to the maximum message length of the queue manager. On other platforms, the value must be greater than or equal to zero, and less than or equal to 4,194,304 bytes.  On IBM MQ for z/OS, the value must be greater than or equal to zero, and less than or equal to 104,857,600 bytes. 	MAXMSGL
Heartbeat interval	<p>Type the length of the heartbeat interval, which can be 0 - 999999. A value of zero means that no heartbeat exchange takes place. Set the value to be less than the value of the Disconnect interval property. The value that is used is the larger of the values specified at the sending side and the receiving side. The heartbeat interval is the time, in seconds, between heartbeat flows passed from the sending MCA when there are no messages on the transmission queue. The heartbeat exchange gives the receiving MCA the opportunity to quiesce the channel.</p>	HBINT

Property	Meaning	MQSC parameter
Maximum instances	<p>This parameter is used on server-connection channels and AMQP channels. Maximum instances specifies the maximum number of simultaneous instances of an individual server-connection channel or AMQP channel.</p> <p>The value can be a number in the range 0 - 999999999. The default value is 999999999</p> <p>A value of zero means that all client access is prevented.</p> <p>If Maximum instances is set to a value which is less than the number of instances of the server-connection channel currently running, then new instances are prevented from starting until sufficient existing instances stop running.</p> <p>V 9.0.0 If a client connects on an AMQP channel with an ID that is already connected (that is, it performs a client-takeover) the takeover will succeed regardless of whether the number of connected clients has reached MAXINST.</p>	MAXINST

Property	Meaning	MQSC parameter
Maximum instances per client	<p>This parameter is used on server-connection channels. Maximum instances per client specifies the maximum number of simultaneous instances of an individual server-connection channel which can be started from a single client. In this context, connections originating from the same remote network address are regarded as coming from the same client.</p> <p>The value can be a number 0 - 999999999. The default value is 999999999</p> <p>A value of zero means that all client access is prevented.</p> <p>Maximum instances differs from Maximum instances per client in that Maximum instances is the maximum amount of connections, but Maximum instances per client is the maximum amount of connections that each client is allowed to connect to the server.</p>	MAXINSTC
Keep alive interval	<p>Type the length of the keep alive interval, 0 - 99999. This property is ignored if the channel uses a transport type other than TCP or SPX. The TCP Keep alive property must be set to Yes on the Channels page of the Queue manager properties.</p> <p> On z/OS queue managers, the Keep alive interval property specifies the keep alive interval for the individual channel.</p> <p>On queue managers on other platforms, the Keep alive interval property is used only if the channel connects to a z/OS queue manager; to use the functionality provided by the Keep alive interval property, set the Keep alive interval property to Auto to use a value based on the negotiated heartbeat interval value.</p>	KAINT

Property	Meaning	MQSC parameter
Sequence number wrap	<p>The sequence number is the count of messages that are sent through the channel. The sequence number increments each time a message is sent through the channel.</p> <p> For z/OS using CICS, type the highest number in the range 1 - 999999999, that the message sequence number reaches before it restarts at 1.</p> <p>For all other platforms, type the highest number in the range 100 - 999999999 that the message sequence number reaches before it restarts at 1.</p> <p>The value must be high enough that the number is not reissued while it is being used by an earlier message. The two ends of the channel must have the same sequence number wrap value when the channel starts; otherwise you get an error.</p>	SEQWRAP
Non-persistent message speed	<p>To specify that nonpersistent messages on a channel are not transferred within a transaction, select Fast. This means that nonpersistent messages become available for retrieval far more quickly than if they are part of a transaction. However, because the nonpersistent messages are not part of a transaction, they might be lost if, for example, the channel stops while the messages are in transit. To prevent this happening, select Normal.</p>	NPMSPEED
Batch size	<p>Type the maximum number of messages to be sent before syncpoint is taken. The messages are always transferred individually but are committed or backed out as a batch. Try the default batch size of 50 and change the value only if you need to.</p>	BATCHSZ

Property	Meaning	MQSC parameter
Message compression	<p>Click Edit to open the Edit Message Compression dialog. Select the message compression techniques that are supported by the channel definition in order of preference. The first technique that is supported by the other end of the channel is used.</p> <p>None means that no message compression is performed; RLE means that message data compression is performed using run-length encoding; ZLIBFAST means that message data compression is performed using the zlib compression technique and a fast compression time is preferred; ZLIBHIGH means that message data compression is performed using the zlib compression technique and a high level of compression is preferred; ANY means that any compression technique that is supported by the queue manager can be used. For more information, see Distributed queueing and clusters.</p>	COMPMSG
Header compression	<p>Click Edit to open the Edit Header Compression dialog. Select the header compression techniques that are supported by the channel definition in order of preference. The first technique that is supported by the other end of the channel is used.</p> <p>None means that no header compression is performed; System means that header compression is performed. For more information, see Distributed queueing and clusters.</p>	COMPHDR
Batch interval	Type the number of milliseconds, 0 - 999999999, during which the channel keeps a batch open even if there are no messages on the transmission queue.	BATCHINT
Batch data limit	Provide the limit in kilobytes, 0 - 999999, of the amount of data that is sent through a channel before taking a sync point. A value of 0 means that no data limit is applied to batches over this channel.	BATCHLIM









Property	Meaning	MQSC parameter
Disconnect interval	Type the number of seconds, 0 - 999999, after the batch ends before the channel closes down. A value of 0 means that the channel does not disconnect.	DISCINT
Data conversion	To specify that the message is converted by the receiving application to the format that is required on the receiving system (this is the typical method), select No ; if the remote queue manager is on a platform that does not support data conversion, select Yes to specify that the message is converted before transmission into the format that is required by the receiving system.	CONVERT
Put authority	This property specifies the type of security processing to be carried out by the Message Channel Agent (MCA) when running an MQPUT command to the target queue or an MQI call. To use the default user ID, click Default ; to use the alternate user ID from the context information that is associated with the message, click Context .	PUTAUT

Property	Meaning	MQSC parameter
Batch heartbeat interval	<p>If the sending channel has had a communication from the receiving channel within the Batch heartbeat interval, the receiving channel is assumed to be still active, otherwise a 'heartbeat' is sent to the receiving channel to check. The sending channel waits for a response from the receiving end of the channel for an interval, based on the number of seconds specified in the channel Heartbeat Interval (HBINT) property.</p> <p>If the receiving end of the channel is not active, the batch can be backed out rather than becoming in-doubt. By backing out the batch, the messages remain available for processing so that they can, for example, be redirected to another channel. Type the number of seconds, 0 - 999999, that the sending end of the channel waits for a response from the receiving end of the channel before assuming that the receiving end of the channel is inactive.</p> <p>A value of 0 means that batch heartbeating is not used. For more information, see “Configuring the channel to reduce the opportunity of being put 'in-doubt’” on page 116.</p>	BATCHHB

Property	Meaning	MQSC parameter
Default channel disposition	<p>When you issue the START CHANNEL command without the channel disposition keyword (CHLDISP), the channel is started using the value of the Default channel disposition (DEFCDISP). The three possible values are:</p> <p>Private. This is the default value. Start as a private channel on the local queue manager.</p> <p>Shared. A receiving channel is shared if it was started in response to an inbound transmission directed to the queue sharing group. A sending channel is shared if its transmission queue has a disposition of SHARED.</p> <p>Fix shared. A sending channel is shared if its transmission queue has a disposition of SHARED and the CONNAME is not blank.</p>	DEFCDISP

Property	Meaning	MQSC parameter
<p>Property control</p> <p>(Only on Sender channels, Server channels, Cluster Sender channels, and Cluster Receiver channels)</p>	<p>This defines what happens to properties of messages that are about to be sent to a V6 or earlier queue manager. The value has to be changed from Compatibility to All to preserve the v6 behavior of propagating the RFH2 to the caller. The possible values are:</p> <p>All means that all properties of the message are included with the message when it is sent to the remote queue manager. The properties, except those in the message descriptor (or extension), are placed in one or more MQRFH2 headers in the message data.</p> <p>Compatibility. This is the default value; it allows applications that expect JMS related properties to be in an MQRFH2 header in the message data to continue to work unmodified.</p> <p>If the message contains a property with a prefix of mcd . , jms . , usr . , or mqext . then all optional message properties (where the Support value is MQPD_SUPPORT_OPTIONAL), except those in the message descriptor (or extension) are placed in one or more MQRFH2 headers in the message data before the message is sent to the remote queue manager. Otherwise all properties of the message, except those in the message descriptor (or extension), are removed from the message before the message is sent to the remote queue manager.</p> <p>If the message contains a property where the Support field of the property descriptor is not set to MQPD_SUPPORT_OPTIONAL then the message are rejected and treated in accordance with its report options. If the message contains one or more properties where the Support field of the property descriptor is set to MQPD_SUPPORT_OPTIONAL but other fields of the property descriptor are set to non-default values, then these properties are removed from the message before the message is sent to the remote queue manager.</p> <p>None means that all properties of the message, except those in the</p>	<p>PROPCTL</p>

Property	Meaning	MQSC parameter
Sharing conversations (Only on Server-connection channels and Client-connection channels)	<p>Specifies the maximum number of conversations that can be shared over a particular TCP/IP client channel instance (socket). The possible values are:</p> <p>0: Specifies no sharing of conversations over a TCP/IP socket. The channel instance runs in a mode before that of IBM WebSphere MQ 7.0, with regard to:</p> <ul style="list-style-type: none"> • Administrator stop-quiesce • Heartbeating • Read ahead <p>1: Specifies no sharing of conversations over a TCP/IP socket. Client heartbeating and read ahead are available, whether in an MQGET call or not, and channel quiescing is more controllable.</p> <p>2 - 999999999: The number of shared conversations. The default value is 10.</p> <p>If the client-connection SHARECNV value does not match the server-connection SHARECNV value, then the lowest value is used.</p>	SHARECNV
Pending reset sequence number	<p>This is the sequence number from an outstanding request and it indicates a user RESET CHANNEL command request is outstanding. A value of zero indicates that there is no outstanding RESET CHANNEL. The value can be in the range 1 - 999999999.</p> <p>When the value of RESETSEQ is 0, the DISPLAY CHANNEL command returns RESETSEQ(NO).</p>	RESETSEQ

Property	Meaning	MQSC parameter
Use dead-letter queue (Not on Client-connection channels, Server-connection channels, or Telemetry channels)	Specifies whether the dead-letter queue is used when messages cannot be delivered by channels. There are two possible values: <ul style="list-style-type: none"> • No means that messages that cannot be delivered by a channel are treated as a failure, and the channel either ends in accordance with the setting of <u>Non-persistent message speed</u>, or discards the messages. • Yes means that if the queue manager <u>Dead-letter queue</u> property provides the name of a Dead Letter Queue, then it is used. Otherwise the behavior is as for No. 	USEDLQ
  Port (On AMQP channels only)	Specifies the port for the AMQP connection. The default port for AMQP 1.0 connections is 5672. If you are already using port 5672, you can specify a different port.	PORT
  Use client ID (On AMQP channels only)	Specifies that the client ID is used for connection on an AMQP channel. Is set to Yes or No.	USECLTID
  AMQP keep alive (On AMQP channels only)	Specifies the keep alive time in milliseconds. If the AMQP client has not sent any frames within the keep alive interval, the connection is closed with a <code>amqp:resource-limit-exceeded</code> AMQP error condition.	AMQPKA
  Topic root (On AMQP channels only)	Specifies the topic root for an AMQP channel. Use this property to ensure that an MQ Light application, when deployed to a queue manager, does not publish or subscribe to messages to or from areas of the topic tree that are being used by other applications. The default value for TPROOT is <code>SYSTEM.BASE.TOPIC</code> . With this default value, the topic string that an AMQP client uses to publish or subscribe has no prefix, and the client can exchange messages with other MQ pub/sub applications.	TPROOT

MCA page

The following table lists the properties that you can set on the **MCA** page of the **Channel properties** dialog. To configure how the Message Channel Agent (MCA) for this channel runs, edit the properties on the **MCA** page.

Property	Meaning	MQSC parameter
MCA user ID	<p>The message channel agent user identifier. If it is nonblank, it is the user identifier that is to be used by the message channel agent for authorization to access IBM MQ resources, including (if PUTAUT is DEF) authorization to put the message to the destination queue for receiver or requester channels.</p> <p>If it is blank, the message channel agent uses its default user identifier. The default user identifier is derived from the user ID that started the receiving channel. The possible values are:</p> <ul style="list-style-type: none">• On z/OS, the user ID assigned to the channel-initiator started task by the z/OS started-procedures table.• For TCP/IP, other than z/OS, the user ID from the inetd.conf entry, or the user that started the listener.• For SNA, other than z/OS, the user ID from the SNA server entry or, in the absence of this the incoming attach request, or the user that started the listener.• For NetBIOS or SPX, the user ID that started the listener. <p>The maximum length of the string is:</p> <ul style="list-style-type: none">• 64 characters on Windows (except channels with CHLTYPE AMQP).• 12 characters for channels with CHLTYPE AMQP on Windows.• 12 characters on platforms other than Windows.	MCAUSER
MCA type	<p>To specify that the message channel agent (MCA) program runs as a thread, select Thread; to specify that the MCA runs as a process, select Process.</p>	MCATYPE

Property	Meaning	MQSC parameter
MCA name	Read-only. You cannot edit this property because the MCA name is reserved and must only be set to blanks.	MCANAME

Exits page

The following table lists the properties that you can set on the **Exits** page of the **Channel properties** dialog. To configure the channel to run user exits, edit the properties on the **Exits** page.

Property	Meaning	MQSC parameter
Send exit name	<p>Click Edit to open the Edit Send Exit Name dialog. Add the names of your send exit programs:</p> <ul style="list-style-type: none"> • On UNIX and Linux, enter the names of one or more exit programs. The maximum total length of all the names together is 999 characters. Use the format <code>libraryname(functionname)</code>, where the maximum number of characters in a name is 128. • On Windows, enter the name of one or more exit programs. The maximum total length of all the names together is 999 characters. Use the format <code>dllname(functionname)</code>, where the maximum number of characters in a name is 128. • On IBM i, enter the names of up to 10 exit programs. Use the format <code>programname libname</code>, where <i>programname</i> occupies the first 10 characters, and <i>libname</i> occupies the second 10 characters. Add spaces at the end of shorter names to bring them up to 10 characters. • On z/OS, enter the names of up to 8 exit programs. Use the load module name, where the maximum number of characters in a name is 8. • On other platforms, you can specify the name of only one send exit program for each channel. 	SENDEXIT

Property	Meaning	MQSC parameter
Send exit user data	<p>Type the data (maximum 32 characters) to be passed to the channel send exit when the send exit program is called:</p> <ul style="list-style-type: none"> • On Windows, UNIX and Linux, type the data for one or more exit programs. Separate the data with commas. The maximum total length of the field is 999 characters. • On IBM i, type up to 10 strings of data, each with a length of 32 characters. The first string of data is passed to the first send exit, the second string is passed to the second exit, and so on. • On z/OS, type up to 8 strings of data, each with a length of 32 characters. The first string of data is passed to the first send exit, the second string is passed to the second exit, and so on. • On other platforms, you can specify only one string of send exit data for each channel. 	SENDDATA

Property	Meaning	MQSC parameter
Receive exit name	<p>Click Edit to open the Edit Receive Exit Name dialog. Add the names of your receive exit programs:</p> <ul style="list-style-type: none"> • On UNIX and Linux, enter the names of one or more exit programs. The maximum total number of characters in all the names together is 999 characters. Use the format <code>libraryname(functionname)</code>, where the maximum number of characters in a string is 128. • On Windows, enter the name of one or more exit programs. Separate the names with commas. The maximum total length of the field is 999 characters. Use the format <code>dllname(functionname)</code>, where the maximum number of characters in a string is 128. • On Windows, type the names of up to 10 exit programs. Separate the names with commas. Use the format <code>programname libname</code>, where <i>programname</i> occupies the first 10 characters, and <i>libname</i> occupies the second 10 characters. Add spaces at the end of shorter names to bring them up to 10 characters. • On z/OS, type the names of up to 8 exit programs. Separate the names with commas. Use the load module name, where the maximum number of characters is 8. • On other platforms, you can specify the name of only one send exit program for each channel. 	RCVEXIT

Property	Meaning	MQSC parameter
Receive exit user data	<p>Type the data (maximum 32 characters) to be passed to the channel receive exit when the receive exit program is called:</p> <ul style="list-style-type: none"> • On Windows, UNIX and Linux, type the data for one or more exit programs. Separate the data with commas. The maximum total length of the field is 999 characters. • On IBM i, type up to 10 strings of data, each with a length of 32 characters. The first string of data is passed to the first receive exit, the second string is passed to the second exit, and so on. • On z/OS, type up to 8 strings of data, each with a length of 32 characters. The first string of data is passed to the first receive exit, the second string is passed to the second exit, and so on. • On other platforms, you can specify only one string of receive exit data for each channel. 	RCVDATA
Security exit name	<p>Type the name of the security exit program:</p> <ul style="list-style-type: none"> • On UNIX and Linux, use the format <code>libraryname(functionname)</code>, where the maximum number of characters in a string is 128. • On IBM i, use the format <code>programname libname</code>, where <i>programname</i> occupies the first 10 characters, and <i>libname</i> occupies the second 10 characters. Add spaces at the end of shorter names to bring them up to 10 characters. • On z/OS, use the load module name, where the maximum number of characters is 8. 	SCYEXIT
Security exit user data	Type the data (maximum 32 characters) to be passed to the channel security exit when the channel security exit is called.	SCYDATA

Property	Meaning	MQSC parameter
Message exit name	<p>Click Edit to open the Edit Message Exit Name dialog. Add the names of your message exit programs:</p> <ul style="list-style-type: none"> • On UNIX and Linux, enter the names of one or more exit programs. The maximum total length of all the names together is 999 characters. Use the format <code>libraryname(functionname)</code>, where the maximum number of characters in a name is 128. • On Windows, enter the name of one or more exit programs. The maximum total length of all the names together is 999 characters. Use the format <code>dllname(functionname)</code>, where the maximum number of characters in a name is 128. • On IBM i, enter the names of up to 10 exit programs. Use the format <code>programname libname</code>, where <i>programname</i> occupies the first 10 characters, and <i>libname</i> occupies the second 10 characters. Add spaces at the end of shorter names to bring them up to 10 characters. • On z/OS, enter the names of up to 8 exit programs. Use the load module name, where the maximum number of characters in a name is 8. • On other platforms, you can specify the name of only one message exit program for each channel. 	MSGEXIT

Property	Meaning	MQSC parameter
Message exit user data	<p>Type the data (maximum 32 characters) to be passed to the channel message exit when the channel message exit program is called:</p> <ul style="list-style-type: none"> • On Windows, UNIX and Linux, type the data for one or more exit programs. Separate the data with commas. The maximum total length of the field is 999 characters. • On IBM i, type up to 10 strings of data, each with a length of 32 characters. The first string of data is passed to the first channel message exit, the second string is passed to the second exit, and so on. • On z/OS, type up to 8 strings of data, each with a length of 32 characters. The first string of data is passed to the first channel message exit, the second string is passed to the second exit, and so on. • On other platforms, you can specify only one string of channel message exit data for each channel. 	MSGDATA

LU6.2 page

The following table lists the properties that you can set on the **LU6.2** page of the **Channel properties** dialog. If the channel uses the LU 6.2 transport protocol, edit the properties on the **LU6.2** page.

Property	Meaning	MQSC parameter
Mode name	Type the LU 6.2 mode name, which is the SNA mode name unless the value of the Connection name property on the General page contains a side-object, in which case leave the Mode name value blank. The maximum length is 8 characters.	MODENAME
TP name	Type the name, or the generic name, of the MCA program that is run at the far end of the link.	TPNAME

Property	Meaning	MQSC parameter
User ID	Type the user identifier that the MCA uses when attempting to initiate a secure LU 6.2 session with a remote MCA. The maximum length is 12 characters; however, only the first 10 characters are used.	USERID
Password	Click Change channel password , then type in the Change Password dialog the password that the MCA uses when it attempts to initiate a secure LU 6.2 session with a remote MCA. The maximum length is 12 characters.	PASSWORD

Retry page

The following table lists the properties that you can set on the **Retry** page of the **Channel properties** dialog. To configure how the channel behaves if the channel cannot connect to the remote queue manager, edit the properties on the **Retry** page.

Property	Meaning	MQSC parameter
Short retry count	Type the maximum number of times, 0 - 999999999 (or, for z/OS using CICS, 1 - 999999999), that the channel can try to connect to a remote queue manager.	SHORTRTY
Short retry interval	Type the approximate interval, in seconds, that the channel must wait before it tries again to connect to the remote queue manager during the short retry count. A value of 0 means that the channel tries again immediately.	SHORTTMR
Long retry count	Type the maximum number of times, 0 - 999999999, that the channel can try to connect to a remote queue manager. The value of this property is used only when the count specified in the Short retry count property has been exhausted and the channel has still not successfully connected to the remote queue manager.	LONGRTY
Long retry interval	Type the approximate interval, in seconds, that the channel must wait before it tries again to connect to the remote queue manager during the long retry count. A value of 0 means that the channel tries again immediately.	LONGTMR

Property	Meaning	MQSC parameter
Keep alive interval	The value of the Keep alive interval property specifies the time-out value of the channel. To base the keepalive value on the value of the negotiated heartbeat interval, select Auto . If the negotiated heartbeat interval is greater than zero, the Keep alive interval is the negotiated heartbeat interval plus 60 seconds; if the negotiated heartbeat interval is zero, the Keep alive interval is zero too. To specify a time-out value, type the number of seconds, 0 - 99999. To disable KeepAlive on this channel, type 0.	KAINT

Message retry page

The following table lists the properties that you can set on the **Message retry** page of the **Channel properties** dialog. To configure how the channel behaves if the channel fails the first time that it tries to put a message on a remote queue, edit the properties on the **Message retry** page.

Property	Meaning	MQSC parameter
Message retry count	Type the number of times, 0 - 999999999, that the channel retries to deliver a message before it decides that it cannot deliver the message to the remote queue. This property controls the action of the MCA only if the Message retry exit name property is blank. If the Message retry exit name property is not blank, the value of the Message retry count property is passed to the exit for the exit's use but the number of times that the channel retries to deliver the message is controlled by the exit, not by the Message retry count property.	MRRTY
Message retry interval	Type the minimum length of time, in milliseconds, that the channel must wait before it can try again to put the message on the remote queue.	MRTMR

Property	Meaning	MQSC parameter
Message retry exit name	Type the name of the channel message retry exit program: <ul style="list-style-type: none"> On UNIX and Linux, use the format <code>libraryname(functionname)</code>, where the maximum number of characters in a string is 128. On IBM i, use the format <code>programname libname</code>, where <i>programname</i> occupies the first 10 characters, and <i>libname</i> occupies the second 10 characters. Add spaces at the end of shorter names to bring them up to 10 characters. On z/OS, use the load module name, where the maximum number of characters is 8. 	MRDATA
Message retry exit user data	Type the data (maximum 32 characters) that is passed to the channel message retry exit when the channel message retry exit is called.	MREXIT

Cluster page

The following table lists the properties that you can set on the **Cluster** page of the **Channel properties** dialog. To share the channel in one or more clusters, edit the properties on the **Cluster** page.

Note: Specify the cluster channel properties on the cluster-receiver channels at the target queue managers. Any properties you specify on the matching cluster-sender channels are likely to be ignored. See [Cluster channels](#) in the IBM MQ online product documentation.

Property	Meaning	MQSC parameter
Not shared in a cluster	By default, this option is selected so that the channel is not shared in any clusters.	Not applicable.
Shared in cluster	To share the channel in a cluster, select this option, then type the name of the cluster. See “Strings in property dialogs” on page 655.	CLUSTER
Shared in a list of clusters	To share the channel in more than one cluster, select this option, then type the name of the namelist object that contains the names of the clusters. See “Strings in property dialogs” on page 655.	CLUSNL

Property	Meaning	MQSC parameter
Network priority	The value of this property indicates the channel priority for the network connection. Type the value, 0 - 9; 0 is the lowest priority. See NETPRTY channel attribute in the IBM MQ online product documentation.	NETPRTY
CLWL channel rank	Type the rank of the channel in the cluster, 0 - 9; 0 is the lowest rank. See CLWLRANK channel attribute in the IBM MQ online product documentation.	CLWLRANK
CLWL channel priority	Type the priority of the channel in the cluster, 0 - 9; 0 is the lowest priority. See CLWLPRTY channel attribute in the IBM MQ online product documentation.	CLWLPRTY
CLWL channel weight	Type the weighting that is applied to the channel so that the proportion of messages that is sent through the channel is controlled. The value must be 1 - 99; 1 is the lowest weighting. See CLWLWGHT channel attribute in the IBM MQ online product documentation.	CLWLWGHT

SSL page

The following table lists the properties that you can set on the **SSL** page of the **Channel properties** dialog. To configure the channel to use SSL security, edit the properties on the **SSL** page.

Property	Meaning	MQSC parameter
CertificateLabel	Certificate label for this channel to use. The label identifies which personal certificate in the key repository is to be sent to the remote peer. If this property is blank, the certificate is determined by the queue manager CertificateLabel property.	CERTLABL
CipherSpec	Type the name (maximum 32 characters) of the CipherSpec for a TLS connection. Both ends of the IBM MQ SSL channel definition must have the same value in the CipherSpec property. For more information, see the SSLCIPH property in DEFINE CHANNEL . The value for this parameter is also used to set the value of the Security Protocol property, which is an output field on the Channel status attributes page.	SSLCIPH

Property	Meaning	MQSC parameter
Authentication of parties initiating connections	To specify that the channel must receive and authenticate an TLS certificate from an TLS client, select Required ; to specify that the channel is not required to receive and authenticate an TLS certificate from an TLS client, select Optional ; if you select Optional and the peer TLS client sends a certificate, the channel authenticates the certificate as normal.	SSLCAUTH
Peer Issuer Name	Certificate issuer Distinguished Name filter. This field contains a Distinguished Name filter which matches the Issuer DN of the remote peer personal certificate. Peer Issuer Name is a key field in the SSL Peer Map, that is, is used to match channel authority records for inbound channel connections.	SSLCERTI
Accept only certificates with Distinguished Names matching these values	Type the value of the Distinguished Name on the certificate from the peer queue manager or client at the other end of the IBM MQ channel. When the channel starts, the value of this property is compared with the Distinguished Name of the certificate.	SSLPEER
Accept only certificates with Distinguished Names matching these values	This channel authentication record maps TLS Distinguished Names (DNs) to MCAUSER values. The SSLPEERMAP parameter must be accompanied by an SSLPEER.	SSLPEERMAP

Load Balancing page

The following table lists the properties that you can set on the **Load Balancing** page of the **Channel properties** dialog.

Property	Meaning	MQSC parameter
Weight	<p>The client channel weighting property is used to specify a weighting to influence which client-connection channel definition is used. The client channel weighting property is used so that client channel definitions can be selected at random based on their weighting when more than one suitable definition is available.</p> <p>When a client issues an MQCONN requesting connection to a queue manager group, by specifying a queue manager name starting with an asterisk, and more than one suitable channel definition is available in the client channel definition table (CCDT), the definition to use is randomly selected based on the weighting, with any applicable CLNTWGHT(0) definitions selected first in alphabetical order. Specify a value in the range 0 - 99. The default is 0. A value of 0 indicates that no load balancing is performed and applicable definitions are selected in alphabetical order. To enable load balancing choose a value in the range 1 - 99 where 1 is the lowest weighting and 99 is the highest. The distribution of messages between two or more channels with non-zero weightings is approximately proportional to the ratio of those weightings.</p>	CLNTWGHT

Property	Meaning	MQSC parameter
Affinity	<p>The channel affinity property is used so client applications that connect multiple times using the same queue manager name can choose whether to use the same client channel definition for each connection. Use this property when multiple applicable channel definitions are available. The possible values are:</p> <p>PREFERRED. This is the default value. The first connection in a process reading a client channel definition table (CCDT) creates a list of applicable definitions based on the client channel weight, with any definitions having a weight of 0 first and in alphabetical order. Each connection in the process attempts to connect using the first definition in the list. If a connection is unsuccessful the next definition is used. Unsuccessful definitions with client channel weight values other than 0 are moved to the end of the list. Definitions with a client channel weight of 0 remain at the start of the list and are selected first for each connection. Each client process with the same host name creates the same list.</p> <p>NONE. The first connection in a process reading a CCDT creates a list of applicable definitions. All connections in a process select an applicable definition based on the client channel weight, with any definitions having a weight of 0 selected first in alphabetical order.</p>	AFFINITY

Statistics page

The following table lists the properties that you can set on the **Statistics** page of the **Channel properties** dialog. To configure the channel to collect monitoring or statistics data, edit the properties on the **Statistics** page.

Property	Meaning	MQSC parameter
Alteration date	Read-only. This is the date on which the queue properties were last altered.	ALTDATE
Alteration time	Read-only. This is the time at which the queue properties were last altered.	ALTTIME

Property	Meaning	MQSC parameter
Channel monitoring	You can configure IBM MQ to collect online monitoring data about the current performance of the channel. To inherit the value of the queue manager's Channel monitoring property (see “Queue manager properties” on page 349), click Queue manager . If the queue manager's Channel monitoring property is None, the queue's Channel monitoring property is ignored. If the queue manager's Channel monitoring property is not None: to override the queue manager settings and prevent data collection for this channel, click Off ; to collect data at a low rate, click Low ; to collect data at a medium rate, click Medium ; to collect data at a high rate, click High .	MONCHL
Channel statistics	You can configure IBM MQ to collect statistics data about the activity of the channel. To inherit the value of the queue manager's Channel statistics property (see Queue Manager properties), click Queue manager . If the queue manager's Channel statistics property is None, the queue's Channel statistics property is ignored. If the queue manager's Channel statistics property is not None, to override the queue manager settings and prevent data collection for this channel, click Off ; to collect data at a low rate, click Low ; to collect data at a medium rate, click Medium ; to collect data at a high rate, click High .	STATCHL

Related tasks

[“Configuring queue managers and objects”](#) on page 51

You can configure many of the attributes of queue managers and their objects from IBM MQ Explorer using the properties dialogs.

Related reference

[“Strings in property dialogs”](#) on page 655

To include certain characters in a string, you must punctuate the string in a particular way.

Listener properties

You can set attributes for all types of listeners. Some attributes are specific to certain types of listener.

The following table lists all the attributes that you can set.

For each attribute, there is a brief description of when you might need to configure the attribute. The tables also give the equivalent MQSC parameter for the ALTER LISTENER and DISPLAY LISTENER commands. For more information about MQSC commands, see [Script \(MQSC\) Commands](#) in the online IBM MQ product documentation.

General page

The following table lists the attributes that you can set on the **General** page of the Listener properties dialog.

Attribute	Meaning	MQSC parameter
Listener name	Read-only. You cannot change the name of the listener after it has been created.	LISTENER
Description	Type a meaningful description of the purpose of the listener. See Entering strings in MQ Explorer .	DESCR
Control	To configure the listener to start and stop when the queue manager starts and stops, click Queue Manager ; to configure the listener to start when the queue manager starts but not stopped when the queue manager is stopped, click Queue Manager Start ; to configure the listener so that it does not start automatically and must be started manually, click Manual .	CONTROL
Transmission protocol	Read-only. This attribute shows the transport protocol that the listener uses. To use a different transport protocol, you must create a new listener object; you cannot change the transport protocol of an existing listener object.	TRPTYPE
Port	Type the port number on which the listener listens for connections.	PORT
IP address	Type the name of the computer on which the listener listens for connections. You can use any of the following formats: IPv4 dotted decimal, IPv6 hexadecimal, or the fully qualified host name, for example <code>joho.hursley.ibm.com</code> . If no value is specified, the listener listens on all available IPv4 and IPv6 addresses.	IPADDR
TP Name	Type the LU 6.2 transaction program name.	TPNAME
Adapter	Type the number of the adapter on which NetBIOS listens. The default value is adapter 0.	ADAPTER

Attribute	Meaning	MQSC parameter
Local Name	Type the NetBIOS local name that the listener uses. The default value is defined by the protocol.	LOCLNAME
Name count	Type the number of names that the listener can use. The default value is defined by the protocol.	NTBNAMES
Session count	Type the number of sessions that the listener can use. The default value is defined by the protocol.	SESSIONS
Command count	Type the number of commands that the listener can use. The default value is defined by the protocol.	COMMANDS
Backlog	Type the maximum number of concurrent connection requests that the listener supports. The default value is defined by the protocol.	BACKLOG
Socket	Type the number of the SPX socket on which the listener listens for connections. The default value is hexadecimal 5E86.	SOCKET
Listener status	Read-only. This attribute shows the current status of the listener, which can be Running, Starting, or Stopping.	STATUS
Alteration date	Read-only. This attribute shows the date on which the listener's attributes were last altered.	ALTDATE
Alteration time	Read-only. This attribute shows the time at which the listener's attributes were last altered.	ALTTIME

General page for z/OS listeners



Z/OS listener properties cannot be altered once the listener has been defined. The properties are set when you add a new z/OS listener.

Attribute	Meaning	MQSC parameter
Listener status	Read-only. This attribute shows the current status of the listener, which can Running, Starting, Retrying, or Stopping.	STATUS

Attribute	Meaning	MQSC parameter
Transmission protocol	Read-only. This attribute shows the transport protocol that the listener uses. To use a different transport protocol, you must create a new listener; you cannot change the transport protocol of an existing listener object.	TRPTYPE
Port number	Read-only. The port number on which the listener listens for connections.	PORT
IP address	Read-only. The name of the computer on which the listener listens for connections.	IPADDR
Inbound	Read-only. This attribute Specifies the disposition of the inbound transmissions that are to be handled. The possible values are: Group or Queue Manager.	INDISP
LU name	Read-only. The LU name of the listener, you can set it when you define the listener.	LUNAME

Related tasks

[“Configuring queue managers and objects” on page 51](#)

You can configure many of the attributes of queue managers and their objects from IBM MQ Explorer using the properties dialogs.

Related reference

[“Strings in property dialogs” on page 655](#)

To include certain characters in a string, you must punctuate the string in a particular way.

Topic properties

An IBM MQ topic is an IBM MQ object that identifies what a publication is about. You can set attributes for topics. Some topic attributes are specific to z/OS topics. Also, there are some attributes that you can alter only while you are creating a topic. You cannot modify these attributes after the IBM MQ topic has been created.

General

The following tables list all the attributes for IBM MQ Topics.

The following table lists the attributes on the **General** page of the IBM MQ **Topic Properties** dialog.

Property	Meaning	MQSC parameter
Topic name	<p>This value cannot be changed after the topic has been created. This parameter is required and cannot contain an empty string.</p> <p>The unique identifier of the administrative topic definition to be created. A maximum of 48 characters are allowed.</p> <p>Topic name must not be the same as any other topic definition defined on the selected queue manager.</p>	TOPNAME
Topic type	This value is read only. This value defines whether the topic is local; Local, or in a cluster; Cluster.	N/A
Topic String	<p>This value cannot be changed after the topic has been created. This parameter is required and cannot contain an empty string.</p> <p>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.</p> <p>Topic String must not be the same as any other topic string already represented by another topic object definition. The maximum length of a topic string is 10 240 characters.</p>	TOPICSTR
Description	<p>This value is a string entered by the administrator. It contains descriptive information about the topic. It must contain only displayable characters. A maximum of 64 characters.</p> <p>If characters are used that are not in the coded character set identifier (CCSID) for the selected queue manager, then they might be translated incorrectly if the information is sent to another queue manager.</p>	DESC
Publish	<p>This property controls whether messages can be published to the topic. The default value is As parent. The 2 other options available are:</p> <p>Allowed which means that messages can be published to the topic by an authorized application.</p> <p>Inhibited which means that messages cannot be published to the topic.</p>	PUB
Subscribe	<p>This property controls whether messages can subscribe to the topic. The default value is As parent. The 2 other options available are:</p> <p>Allowed which means that subscriptions can me made to the topic by an authorized application.</p> <p>Inhibited which means that applications cannot subscribe to the topic.</p>	SUB
Durable subscriptions	<p>This property controls whether the topic permits durable subscriptions to be made. The default value is As parent. The 2 other options available are:</p> <p>Allowed which means that durable subscriptions can me made to the topic by an application.</p> <p>Inhibited which means that durable subscriptions cannot be made to the topic by an application.</p>	DURSUB

Property	Meaning	MQSC parameter
Default priority	<p>The default priority of messages published to the topic. The default value is <code>As parent</code>.</p> <p>The default priority can be set from 0 (the lowest priority) to 9 (the highest priority)</p>	DEFPRTY
Default persistence	<p>The default persistence of a new topic is <code>As parent</code>. Select <code>Persistent</code> to specify that messages created by applications that use <code>MQPER_PERSISTENCE_AS_Q_DEF</code> become persistent. Select <code>Not Persistent</code> to specify that messages created by applications that use <code>MQPER_PERSISTENCE_AS_Q_DEF</code> become nonpersistent.</p>	DEFPSIST
Model durable queue	<p>This value is a string entered by the administrator. It contains the name of the model queue used for durable subscriptions that request that the queue manager manages the destination of its publications.</p> <p>A maximum of 48 characters are allowed for the name.</p> <p>If this field is blank, then it is treated as <code>As parent</code></p> <p>If you are specifying a model queue for a clustered topic, you must ensure that the queue is defined on every queue manager in the cluster where a durable subscription using this topic can be made.</p> <p>The dynamic queue created from this model has a prefix of <code>SYSTEM.MANAGED.DURABLE</code></p>	MDURMDL
Model non-durable queue	<p>This value is a string entered by the administrator. It contains the name of the model queue used for nondurable subscriptions that request that the queue manager manages the destination of its publications.</p> <p>A maximum of 48 characters are allowed for the name.</p> <p>If this field is blank, then it is treated as <code>As parent</code></p> <p>If you are specifying a model queue for a clustered topic, you must ensure that the queue is defined on every queue manager in the cluster where a non-durable subscription using this topic can be made.</p> <p>The dynamic queue created from this model has a prefix of <code>SYSTEM.MANAGED.NDURABLE</code></p>	MNDURMDL
QSG disposition	<p>The queue sharing group disposition of the topic. You can set the queue sharing group disposition to one of three values:</p> <ul style="list-style-type: none"> • <code>Queue manager</code> means that the object definition is available only to the queue manager that hosts it. • <code>Group</code> means that the object definition is stored on the shared repository and each queue manager in the queue-sharing group has a copy of the definition. • <code>Copy</code> means that the object definition is the queue manager's copy of a definition in the shared repository. <p>The field is displayed as read only when you are displaying the properties of a topic.</p>	QSGDISP

Property	Meaning	MQSC parameter
Default put response type	<p>The default response type for message puts. The default value is As parent. The 2 other options available are:</p> <p>Synchronous which means the response is put synchronously.</p> <p>Asynchronous which means the response is put asynchronously.</p>	DEFRESP
Non-persistent message delivery	<p>The delivery method for non-persistent messages published to this topic. The 4 options are:</p> <p>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. This is the default supplied with IBM MQ, but your installation might have changed it.</p> <p>To all available subscribers 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.</p> <p>To all durable subscribers 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 calls fails.</p> <p>To all subscribers 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 call fails.</p>	NPMMSGDLV
Persistent message delivery	<p>The delivery method for persistent messages published to this topic. The four options are:</p> <p>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. This is the default supplied with IBM MQ, but your installation might have changed it.</p> <p>To all available subscribers 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.</p> <p>To all durable subscribers 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 calls fails.</p> <p>To all subscribers 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 call fails.</p>	PMSGDLV

Property	Meaning	MQSC parameter
Wildcard operation	<p>This value controls the behavior of wildcard subscriptions with respect to the topic. The 2 values are:</p> <p>Block. Subscriptions made to a wildcard topic less specific than the topic string for this topic object, do not receive publications made to this topic or to topic strings more specific than this topic.</p> <p>Passthrough. Subscriptions made to a wildcard topic less specific than the topic string for this topic object receive publications made to this topic and to topic strings more specific than this topic. This is the default value.</p>	WILDCARD
Use dead-letter queue	<p>Specifies whether the dead-letter queue is used when publication messages cannot be delivered to their correct subscriber queue. There are three possible values:</p> <ul style="list-style-type: none"> • No means that publication messages that cannot be delivered to their correct subscriber queue are treated as a failure to put the message, and the application's MQPUT to a topic fails in accordance with the settings of <u>Non-persistent message delivery</u> and <u>Persistent message delivery</u>. • Yes means that if the queue manager Dead-letter queue attribute provides the name of a Dead Letter Queue, then it is used. Otherwise the behavior is as for No. • As parent means that the decision to use the Dead Letter Queue is based on the setting of the closest administrative topic object in the topic tree. This is the default supplied with IBM MQ but your installation might have changed it. 	USEDLQ
Custom	<p>The Custom parameter is reserved for the configuration of new features before separate attributes have been introduced. The possible values are a list of zero or more attributes-value pairs, in MQSC-style syntax, separated by at least 1 space.</p> <p>The attribute names and values are case-sensitive, and must be specified in uppercase. The values can contain spaces and parentheses, but must not contain single quotation marks. Examples of valid syntax are:</p> <ul style="list-style-type: none"> • CUSTOM(' ') • CUSTOM('A(B)') • CUSTOM('C(D) E(F)') • CUSTOM('G(5000) H(9.20.4.6(1415))') <p>The queue manager parses the value, but if the string cannot be parsed according to these rules, or if it contains attributes or values that are not recognized, the queue manager ignores the errors.</p>	CUSTOM

Distributed Pub/Sub

The following table lists the attributes on the **Distributed Pub/Sub** page of the IBM MQ **Topic Properties** dialog.

Property	Meaning	MQSC parameter
Proxy subscription behavior	<p>Proxy-subscriptions are associated with the queue manager name that created them. Publications are only forwarded to directly connected queue managers if a proxy-subscription exists that includes the publication topic. The two options for this value are:</p> <p>Force. This forces the sending of a wild-carded proxy-subscription for the topic string associated with this topic object from every queue manager in the cluster to every other queue manager in the pub/sub topology, regardless of whether any local subscriptions have been made. Once this forced proxy subscription has been propagated throughout the topology, any new subscriptions immediately receive any publications from other connected queue manager without suffering latency, although all publications are propagated to all other queue managers in the cluster regardless of whether a subscription has requested them or not.</p> <p>Setting this value at a given level in the topic tree also prevents proxy subscriptions being generated for individual topic strings at subsequent levels in the topic tree, reducing the proxy subscription overhead.</p> <p>First use. For each unique topic string at or beneath this topic object, a proxy subscription is asynchronously sent to all neighboring queue managers in the following scenarios:</p> <ul style="list-style-type: none"> • When a local subscription is created. • When a proxy subscription is received that must be propagated to further directly connected queue managers. 	PROXYSUB

Property	Meaning	MQSC parameter
Publication scope	<p>The scope of publications can be controlled administratively using the PUBSCOPE topic attribute. The attribute can be set to one of the following 3 values:</p> <ul style="list-style-type: none"> • As parent. This is the default value. The publication scope is set to the same value as the parent queue manager. • Queue manager. The publication is only delivered to local subscribers. • All. The publication is delivered to local subscribers and remote subscribers by directly connected queue managers. 	PUBSCOPE
Subscription scope	<p>The scope of subscriptions can be controlled administratively using the SUBSCOPE topic attribute. The attribute can be set to one of the following 3 values:</p> <ul style="list-style-type: none"> • As parent. This is the default value. The subscription scope is set to the same value as the parent queue manager. • Queue manager. The subscription receives only local publications, and proxy subscriptions are not propagated to remote queue managers. • All. A proxy subscription is propagated to remote queue managers, and the subscriber receives local and remote publications. 	SUBSCOPE
Multicast	<p>This attribute controls whether the topic is considered to be transmittable with multicast or not. There are four possible values:</p> <p>As parent. The multicast attribute of the topic is inherited from the parent.</p> <p>Disabled. No multicast traffic is allowed at this node.</p> <p>Enabled. Multicast traffic is allowed at this node.</p> <p>Only. Only subscriptions from a multicast capable client are allowed.</p>	MCAST

Property	Meaning	MQSC parameter
Communication information	The communication information object name. As there are more than one topics in the tree which require the same multicast transmission attributes, consider having these attributes in a separate object which can be referenced.	COMMINFO

Cluster

The following table lists the attributes on the **Cluster** page of the IBM MQ **Topic Properties** dialog.

Property	Meaning	MQSC parameter
Cluster name	To make a topic a cluster topic, you configure this property. When you do this, any topic used by a publisher or subscriber at that point or in the topic tree is shared across all queue managers in the cluster, and messages published to a clustered branch of the topic tree are automatically routed to subscriptions on other queue managers in the cluster.	CLUSTER

Property	Meaning	MQSC parameter
Cluster object state	<p>The current state of this topic object in this cluster. The values can be as follows:</p> <p>ACTIVE The cluster topic is correctly configured and being adhered to by this queue manager.</p> <p>PENDING Only seen by a hosting queue manager, this state is reported when the topic has been created but the full repository has not yet propagated it to the cluster. This might be because the host queue manager is not connected to a full repository, or because the full repository has deemed the topic to be invalid.</p> <p>INVALID This clustered topic definition conflicts with an earlier definition in the cluster and is therefore not currently active.</p> <p>ERROR An error has occurred with respect to this topic object.</p> <p>This parameter is typically used to aid diagnosis when multiple definitions of the same clustered topic are defined on different queue managers, and the definitions are not identical.</p>	CLSTATE

Property	Meaning	MQSC parameter
Cluster route	<p>The routing behavior to use for topics in the cluster defined by the CLUSTER parameter. There are two possible values:</p> <p>DIRECT When you configure a direct routed clustered topic on a queue manager, all queue managers in the cluster become aware of all other queue managers in the cluster. When performing publish and subscribe operations, each queue manager then connects directly to all the others.</p> <p>TOPICHOST When you use topic host routing, all queue managers in the cluster become aware of the cluster queue managers that host the routed topic definitions. When performing publish and subscribe operations, queue managers in the cluster connect only to these topic host queue managers, and not directly to each other. The topic host queue managers are responsible for routing publications from queue managers on which publications are published to queue managers with matching subscriptions.</p>	CLROUTE

Statistics

The following table lists the attributes on the **Statistics** page of the IBM MQ **Topic Properties** dialog.

Property	Meaning	MQSC parameter
Alteration date	<p>This value cannot be changed, it is provided for information purposes only.</p> <p>This is the date on which the topic attributes were last altered.</p>	ALTDATE
Alteration time	<p>This value cannot be changed, it is provided for information purposes only.</p> <p>This is the time at which the topic attributes were last altered.</p>	ALTTIME

Related tasks

[“Creating and configuring queue managers and objects” on page 27](#)

You can create, configure, and delete queue managers and objects in IBM MQ Explorer by using the Navigator view and Content view.

[“Comparing the properties of two objects” on page 53](#)

You can compare the properties of an object with another object of the same type; for example, compare a queue with another queue, a topic with another topic, or a channel with another channel.

Service properties

You can configure attributes for custom service objects in the Service properties dialog.

The following table lists all the attributes that you can set.

For each attribute, there is a brief description of when you might need to configure the attribute. The tables also give the equivalent MQSC parameter for the ALTER SERVICE and DISPLAY SERVICE commands. For more information about MQSC commands, see [Script \(MQSC\) Commands](#) in the online IBM MQ product documentation.

General page

The following table lists the attributes that you can set on the **General** page of the Service properties dialog.

Attribute	Meaning	MQSC parameter
Service name	Read-only. This attribute shows the name of the service.	SERVICE
Description	Type a meaningful description of the purpose of the service. See Entering strings in IBM MQ Explorer .	DESCR
Service control	To configure the service to start and stop automatically when the queue manager starts and stops, click Queue Manager ; to configure the service to start automatically when the queue manager starts but not to stop when the queue manager stops, click Queue Manager Start ; to configure the service so that you must manually start and stop it, click Manual .	CONTROL
Start command	Type the fully-qualified path to the start command program that runs when the service starts; for example, C:\Program Files\IBM\WebSphere MQ\bin\runmqchi.exe	STARTCMD
Start args	Type any arguments that are to be passed to the program when it starts.	STARTARG
Stop command	Type the fully-qualified path to the stop command program that runs when the service stops.	STOPCMD

Attribute	Meaning	MQSC parameter
Stop args	Type any arguments that are to be passed to the program when it stops.	STOPARG
StdOut	Type the path to the file to which the standard output of the service program is written. If the file does not exist when the service program is started, the file is created; if the file already exists, the new standard output is appended to the existing file. If the value of this attribute is blank, the standard output is discarded.	STDOUT
StdErr	Type the path to the file to which the standard error of the service program is written. If the file does not exist when the service program is started, the file is created; if the file already exists, the new standard error is appended to the existing file. If the value of this attribute is blank, the standard error is discarded.	STDERR
Service type	To enable only one instance of the service to run at a time, click Server ; to enable multiple instances of the service to run at a time, click Command .	SERVTYPE
Service status	Read-only. This attribute shows the current status of the service.	STATUS

Related tasks

[“Configuring queue managers and objects” on page 51](#)

You can configure many of the attributes of queue managers and their objects from IBM MQ Explorer using the properties dialogs.

Related reference

[“Strings in property dialogs” on page 655](#)

To include certain characters in a string, you must punctuate the string in a particular way.

IBM MQ service definition properties

You can set properties and attributes for service definitions while creating a new service definition, or when editing an existing service definition.

Some attributes will only be available when the service definition is of a specific binding type or message exchange pattern. Follow these links to view the properties for the individual property pages of a service definition:

- [General](#)
- [Operation](#)
- [Input destination](#)
- [Input message schema](#)

- [Input message header](#)
- [Output destination](#)
- [Output message schema](#)
- [Output message header](#)

For each attribute, there is a brief description of when you might need to configure the attribute.

General page

The following table lists the attributes you can set on the **General** page of the **Service definition properties** dialog.

Attribute	Description
Namespace	Specifies the namespace for the service. There is already a temporary default value assigned.
Name	A unique name for the new Service Definition. The service definition name is not case sensitive, but a mixed-case service definition name is retained.
Message exchange pattern	The Message Exchange Pattern describes the direction of messages sent and received during the invocation of a service. There are two possible selections: <ul style="list-style-type: none"> • One -Way means that a message is sent one way only. • Request -Response means that a message is sent and a response is received.
Binding type	Specifies the version of the IBM MQ service definition specification.
Soap version	Available for SOAP binding types only. Specifies the type and version of the SOAP payload. There are two possible selections: <ul style="list-style-type: none"> • SOAP 1.1 for XML SOAP 1.1 messages. This is the default value. • SOAP 1.2 for XML SOAP 1.2 messages
Comment	Specify a comment to annotate the service in the WSDL file.

Operation page

The following table lists the attributes you can set on the **Operation** page of the **Service definition properties** dialog. Each service definition has only 1 operation.

Attribute	Description
Operation name	Specifies the name of the operation. This property must have a value to create a service definition.

Attribute	Description
Action	<p>Used by the service provider to dispatch service requests. For example: Allowing multiple services to be deployed using a single destination, enabling a service provider to dispatch the requests arriving on the destination appropriately.</p> <p>When the binding type is <i>MQ</i>, Action specifies targetAction.</p> <p>When the binding type is <i>SOAP/MQ</i>, Action specifies SOAPAction.</p>
Comment	Specify a comment to annotate the operation in the WSDL file.

Input destination page

The following table lists the attributes you can set on the **Input Destination** page of the **Service definition properties** dialog. The input pages define the details of the message that the service expects, and the destination where it will be retrieved from.

Attribute	Description
Input destination name	<p>Specifies the name of the destination queue or the destination topic to which the request is sent, for example:</p> <p>The queue-dest or topic-dest particle of an IBM MQ IRI, such as:</p> <pre>msg/queue/INS.QUOTE.REPLY</pre>
Destination queue manager name	Specifies the name of the destination queue manager.
Connection queue manager	Specifies the name of the queue manager to which the requesting service connects to. This corresponds to the QmgrName parameter used on the MQCONN() and MQCONNX() calls.
Client connection properties	The client connection properties specify detailed bindings which can include information about how a service requester binds to a specific machine or channel. Being able to specify client-bindings and channel names is useful in some circumstances, but over-specifying the service might be restrictive however. A solution to this problem is to minimize the amount of binding information incorporated into a service definition and allow underlying infrastructure or IBM MQ to route messages where possible.

Attribute	Description
Channel table name	<p>Specifies the name of the client channel table file which is used to identify the channel connection.</p> <ul style="list-style-type: none"> • If Channel table name is not specified, then Channel table library is ignored. • If either of the MQSERVER or MQCHLTAB environment variables are set in the environment where the client application is running, then Channel table name is ignored.
Channel table library	<p>Specifies the path to the client channel table.</p> <ul style="list-style-type: none"> • If either MQSERVER or MQCHLLIB environment variables are set in the environment where the client application is running, then Channel table library is ignored. • If Channel table name is not specified, then Channel table library is ignored.
Client channel connection name	<p>Specifies the connection string used when a service requester makes an IBM MQ MQI client-binding connection. For TCP/IP, the connection is in the form of a host name followed by a port number, for example:</p> <div data-bbox="824 940 1466 1018" style="background-color: #f0f0f0; padding: 5px; margin: 10px 0;"> OS2R0G3(1822) </div> <p>If the port number is not specified, a default value of 1414 is used.</p> <ul style="list-style-type: none"> • If the Client channel connection name is specified, then the Client channel name and Client channel transport type must also be specified. • If either MQSERVER or MQCHLTAB environment variable are set in the environment where the client application is running, then Client channel connection name is ignored.
Client channel name	<p>Specifies the channel used when an IBM MQ service requester make an IBM MQ MQI client-binding connection.</p> <ul style="list-style-type: none"> • If the Client channel connection name is specified, then the Client channel name and Client channel transport type must also be specified. • If either MQSERVER or MQCHLTAB environment variable are set in the environment where the client application is running, then Client channel name is ignored.

Attribute	Description
Client channel transport type	<p>Specifies the transport type to be used when an IBM MQ service requester makes an IBM MQ MQI client-binding connection.</p> <ul style="list-style-type: none"> • If the Client channel connection name is specified, then the Client channel name and Client channel transport type must also be specified. • If either MQSERVER or MQCHLTAB environment variable are set in the environment where the client application is running, then Transport type is ignored. <p>There are two different selectable values:</p> <ul style="list-style-type: none"> • TCP. Used to specify the TCP/IP transport protocol. This is the default value. • LU62. Used to specify the LU6.2 transport protocol.

Input message schema page

The following table lists the attributes you can set on the **Input message schema** page of the **Service definition properties** dialog. They allow the schema for the message payload to be defined.

Attribute	Description
Inbound data type	Specifies the expected inbound data type. For simple types, this can be modelled using the built-in XML xsd types such as xsd:string or xsd:int. For more complex types, a data type can be imported from an external file by specifying the Import schema file and Import namespace for the data type.
Import schema file	Specifies the schema file to be imported.
Import namespace	Specifies the namespace to be imported.

Input message header page

The following table lists the attributes you can set on the **Input message header** page of the **Service definition properties** dialog. The input pages define the details of the message that the service expects, and the destination where it will be retrieved from. Some of the properties only apply to service definitions of an MQ binding type.

Attribute	Description
CCSID	Specifies the Coded Character Set ID which corresponds to the <i>CodedCharSetId</i> field in the MQMD structure. If this value is not specified, then the service requester and service provider uses the value which corresponds to the character set of the message data.

Attribute	Description
Format	<p>Specifies the format name of the message data. This property corresponds to the <i>MQRFH2</i> format field, or the <i>MQMD</i> format field if there is no <i>MQRFH2</i> is present. The value must be a character string between 0 and 8 characters long consisting of the A-Z and 0-9 characters.</p> <p>For non-SOAP messages, the <i>Format</i> can be set to any value according to the guidelines in the Format field.</p> <p>For SOAP messages, this value is already defined by the value set in SOAP Version in the “General page” on page 470.</p>
User properties	<p>Specifies the user defined data carried in IBM MQ service messages. Values must be specified in the format allowed for RFH2 folder elements; in a series of triplets encoded using an XML-like syntax such as:</p> <pre data-bbox="834 762 1243 787"><name dt="datatype">value</name></pre> <p>The dt="datatype" element is optional, and if it is omitted, it is treated as a string, allowing elements to be specified as:</p> <pre data-bbox="834 961 1065 987"><name>value</name></pre> <p>For example:</p> <pre data-bbox="834 1098 1370 1144"><myprop1>value1</myProp1><myprop2>value2</myProp2><myprop3 dt="i4">99</myProp3></pre> <p>For further information on the allowed data types and formatting, see NameValueData (MQCHARn) in the online product documentation. It is not recommended to include security sensitive properties such as UserId or passwords.</p>

Attribute	Description
Message type	<p>Specifies the type of message that is sent. This property corresponds to the <i>MsgType</i> in the MQMD structure. The five possible values are:</p> <ul style="list-style-type: none"> • Unspecified which means that no value is set, therefore the value is taken from the value of the Message exchange pattern. This is the default value. • Request which means that the message is one that requires a reply. This value indicate that the service uses a request-response message exchange pattern. • Reply which means that the message is a reply to a request. • Report which means that the message is a report. • Datagram which means that the service is a one-way message exchange and there will not be a reply. <p>If no value is specifies, the value is set according to the Message Exchange Pattern.</p>
Persistence	<p>Specifies whether the message is persistent or not, and it corresponds to the <i>Persistence</i> field in the MQMD structure. The three possible values are:</p> <ul style="list-style-type: none"> • Not persistent which means that the messages are not persistent. • Persistent which means that the messages are persistent. • Queue default which means that the queue manager determines the message persistence from the definition of the destination that the message is put to. This is the default value.
Message ID	<p>Specifies the Message identifier which corresponds to the <i>MsgId</i> field in the MQMD structure.</p> <p>The Message ID allows certain specialized IBM MQ applications to be described as services (for example: applications that share an input queue and select the messages intended for them based on a pre-defined msgId value). Predefined <i>msgIds</i> in service definitions might lead to problems such as when a request-response Message Exchange Pattern return the request's <i>msgId</i>.</p> <p>The Message Id can be either a character string or a binary value. Binary values must be a string of up to 24 pairs of two-character hexadecimal values.</p> <p>Click Edit to open a dialog and enter a value as text or bytes</p>

Attribute	Description
Correlation ID	<p>Specifies the Correlation ID corresponds to the CorrelId field in the MQMD structure. The Correlation ID can be either a character string or a binary value. Binary values must be a string of up to 24 pairs of two-character hexadecimal values.</p> <p>Click Edit to open a dialog and enter a value as text or bytes</p>
Expiry	<p>Specifies the message lifetime. It must be a signed integer, and it is measured in tenths of a second. The Expiry range is from 1 to 2 147 483 647.</p> <p>A special value of Unlimited is used to indicate that the message does not expire. The value of -1 is written to the WSDL file.</p> <p>A value of Unspecified means that no value is written to the WSDL file. This is the default value.</p>
Priority	<p>Specifies the priority associated with the message and corresponds to the <i>priority</i> field in the MQMD structure. It must be specified as an integer between 0 and 9, with 0 as the lowest priority and 9 as the highest priority.</p> <p>A special value of Unlimited is used to indicate that the message priority is taken from the definition of the first queue that the message is put to. The value of -1 is written to the WSDL file.</p> <p>A value of Unspecified means that no value is written to the WSDL file. This is the default value.</p>
Encoding	<p>Specifies the numeric encoding of message data, which corresponds to the <i>Encoding</i> field in the MQMD structure:</p> <ul style="list-style-type: none"> • Integer allows you to select Normal or Reversed • Decimal allows you to select Normal or Reversed • Float allows you to select Normal, Reversed, or S390 • Mnemonic specifies the 3 character mnemonic based on the other values selected. R = Reversed, N = Normal, and 3 = S390. • Value specifies the numeric value of the selections and mnemonic.

Attribute	Description
Report options	<p>Specifies how the message and correlation identifiers in the reply message or fault message are set by the service provider. This property corresponds to the Report field in the MQMD structure. There are four possible values:</p> <ul style="list-style-type: none"> • New message ID indicates that if a report or reply is generated as a result of this message, a new <i>msgId</i> is generated for the report or reply message. • Pass Message ID indicates that if a report or reply is generated as a result of this message, then the <i>msgId</i> of this message is copied to the <i>msgId</i> of the report or reply message. • copy Message ID to Correlation ID indicates that if a report or reply is generated as a result of this message, then the <i>msgId</i> of this message is copied to the <i>correlId</i> of the report or reply message. • Pass Correlation ID indicates that if a report or reply is generated as a result of this message, then the <i>correlId</i> of this message is copied to the correlId of the report or reply message.

Output destination page

The following table lists the attributes you can set on the **Output destination** page of the **Service definition properties** dialog. The output pages define the details of the message that the service will send in response to the input message, and the destination where it will be put. The output destination name must be prefixed with 'msg/queue/' for queues, or 'msg/topic/' for topics.

Attribute	Description
Output destination name	<p>Specifies the name of the destination queue or the destination topic to which the response message is sent, and corresponds to the ReplyToQ and ReplyToQMgr fields of the MQMD structure. The Destination Name must take the form of either the queue-dest or topic-dest particle of an IBM MQ URI, such as:</p> <pre>msg/queue/INS.QUOTE.REPLY</pre>
Destination queue manager name	Specifies the name of the destination queue manager.
Connection queue manager	Specifies the name of the queue manager to which the requesting service connects to. This corresponds to the QmgrName parameter used on the MQCONN() and MQCONNX() calls.

Attribute	Description
Client connection properties	<p>The client connection properties specify detailed bindings which can include information about how a service requester binds to a specific machine or channel. Being able to specify client-bindings and channel names is useful in some circumstances, but over-specifying the service might be restrictive however. A solution to this problem is to minimize the amount of binding information incorporated into a service definition and allow underlying infrastructure or IBM MQ to route messages where possible.</p>
Channel table name	<p>Specifies the name of the client channel table file which is used to identify the channel connection.</p> <ul style="list-style-type: none"> • If Channel table name is not specified, then Channel table library is ignored. • If either of the MQSERVER or MQCHLTAB environment variables are set in the environment where the client application is running, then Channel table name is ignored.
Channel table library	<p>Specifies the path to the client channel table.</p> <ul style="list-style-type: none"> • If either MQSERVER or MQCHLLIB environment variables are set in the environment where the client application is running, then Channel table library is ignored. • If Channel table name is not specified, then Channel table library is ignored.
Client channel name	<p>Specifies the connection string used when a service requester makes an IBM MQ MQI client-binding connection. For TCP/IP, the connection is in the form of a host name followed by a port number, for example:</p> <div data-bbox="824 1283 1471 1356" style="background-color: #f0f0f0; padding: 5px; margin: 10px 0;"> OS2R0G3(1822) </div> <p>If the port number is not specified, a default value of 1414 is used.</p> <ul style="list-style-type: none"> • If the Client channel connection name is specified, then the Client channel name and Client channel transport type must also be specified. • If either MQSERVER or MQCHLTAB environment variable are set in the environment where the client application is running, then Client channel connection name is ignored.

Attribute	Description
Client channel connection name	<p>Specifies the channel used when an IBM MQ service requester make an IBM MQ MQI client-binding connection.</p> <ul style="list-style-type: none"> • If the Client channel connection name is specified, then the Client channel name and Client channel transport type must also be specified. • If either MQSERVER or MQCHLTAB environment variable are set in the environment where the client application is running, then Client channel name is ignored.
Client channel transport type	<p>Specifies the transport type to be used when an IBM MQ service requester makes an IBM MQ MQI client-binding connection.</p> <ul style="list-style-type: none"> • If the Client channel connection name is specified, then the Client channel name and Client channel transport type must also be specified. • If either MQSERVER or MQCHLTAB environment variable are set in the environment where the client application is running, then Transport type is ignored. <p>There are two different selectable values:</p> <ul style="list-style-type: none"> • TCP. Used to specify the TCP/IP transport protocol. This is the default value. • LU62. Used to specify the LU6.2 transport protocol.

Output message schema page

The following table lists the attributes you can set on the **Output message schema** page of the **Service definition properties** dialog. They allow the schema for the message payload to be defined.

Attribute	Description
Outbound data type	Specifies the expected outbound data type
Import Schema File	Specifies the schema file to be imported
Import Namespace	Specifies the namespace to be imported

Output message header page

The following table lists the attributes you can set on the **Output message header** page of the **Service definition properties** dialog. The output pages define the details of the message that the service will send in response to the input message, and the destination where it will be put. Some of the properties only apply to service definitions of an MQ binding type.

Attribute	Description
CCSID	Specifies the Coded Character Set ID which corresponds to the <i>CodedCharSetId</i> field in the MQMD structure. If this value is not specified, then the service requester and service provider uses the value which corresponds to the character set of the message data.
Format	<p>Specifies the format name of the message data. This property corresponds to the <i>MQRFH2</i> format field, or the <i>MQMD</i> format field if there is no <i>MQRFH2</i> is present. The value must be a character string between 0 and 8 characters long consisting of the A-Z and 0-9 characters.</p> <p>For non-SOAP messages, the <i>Format</i> can be set to any value according to the guidelines in the Format field.</p> <p>For SOAP messages, this value is already defined by the value set in SOAP Version in the “General page” on page 470.</p>
User properties	<p>Specifies the user defined data carried in IBM MQ service messages. Values must be specified in the format allowed for RFH2 folder elements; in a series of triplets encoded using an XML-like syntax such as:</p> <pre data-bbox="834 972 1243 997"><name dt="datatype">value</name></pre> <p>The dt="datatype" element is optional, and if it is omitted, it is treated as a string, allowing elements to be specified as:</p> <pre data-bbox="834 1171 1065 1197"><name>value</name></pre> <p>For example:</p> <pre data-bbox="834 1308 1370 1354"><myprop1>value1</myProp1><myprop2>value2</myProp2><myprop3 dt="i4">99</myProp3></pre> <p>For further information on the allowed data types and formatting, see NameValueData (MQCHARn). It is not recommended to include security sensitive properties such as UserId or passwords.</p>

Attribute	Description
Message type	<p>Specifies the type of message that is sent. This property corresponds to the <i>MsgType</i> in the MQMD structure. The five possible values are:</p> <ul style="list-style-type: none"> • Unspecified which means that no value is set, therefore the value is taken from the value of the Message exchange pattern. This is the default value. • Request which means that the message is one that requires a reply. This value indicate that the service uses a request-response message exchange pattern. • Replay which means that the message is a reply to a request. • Report which means that the message is a report. • Datagram which means that the service is a one-way message exchange and there will not be a reply. <p>If no value is specifies, the value is set according to the Message Exchange Pattern.</p>
Persistence	<p>Specifies whether the message is persistent or not, and it corresponds to the <i>Persistence</i> field in the MQMD structure. The three possible values are:</p> <ul style="list-style-type: none"> • Not persistent which means that the messages are not persistent. • Persistent which means that the messages are persistent. • Queue default which means that the queue manager determines the message persistence from the definition of the destination that the message is put to. This is the default value.
Message ID	<p>Specifies the Message identifier which corresponds to the <i>MsgId</i> field in the MQMD structure.</p> <p>The Message ID allows certain specialized IBM MQ applications to be described as services (for example: applications that share an input queue and select the messages intended for them based on a pre-defined msgId value). Predefined <i>msgIds</i> in service definitions might lead to problems such as when a request-response Message Exchange Pattern return the request's <i>msgId</i>.</p> <p>The Message Id can be either a character string or a binary value. Binary values must be a string of up to 24 pairs of two-character hexadecimal values.</p> <p>Click Edit to open a dialog and enter a value as text or bytes</p>

Attribute	Description
Correlation ID	<p>Specifies the Correlation ID corresponds to the CorrelId field in the MQMD structure. The Correlation ID can be either a character string or a binary value. Binary values must be a string of up to 24 pairs of two-character hexadecimal values.</p> <p>Click Edit to open a dialog and enter a value as text or bytes</p>
Expiry	<p>Specifies the message lifetime. It must be a signed integer, and it is measured in tenths of a second. The Expiry range is from 1 to 2 147 483 647.</p> <p>A special value of Unlimited is used to indicate that the message does not expire. The value of -1 is written to the WSDL file.</p> <p>A value of Unspecified means that no value is written to the WSDL file. This is the default value.</p>
Priority	<p>Specifies the priority associated with the message and corresponds to the priority field in the MQMD structure. It must be specified as an integer between 0 and 9, with 0 as the lowest priority and 9 as the highest priority.</p> <p>A special value of Unlimited is used to indicate that the message priority is taken from the definition of the first queue that the message is put to. The value of -1 is written to the WSDL file.</p> <p>A value of Unspecified means that no value is written to the WSDL file. This is the default value.</p>
Encoding	<p>Specifies the numeric encoding of message data, which corresponds to the <i>Encoding</i> field in the MQMD structure:</p> <ul style="list-style-type: none"> • Integer allows you to select Normal or Reversed • Decimal allows you to select Normal or Reversed • Float allows you to select Normal, Reversed, or S390 • Mnemonic specifies the 3 character mnemonic based on the other values selected. R = Reversed, N = Normal, and 3 = S390. • Value specifies the numeric value of the selections and mnemonic.

Attribute	Description
Report options	<p>Specifies how the message and correlation identifiers in the reply message or fault message are set by the service provider. This property corresponds to the Report field in the MQMD structure. There are four possible values:</p> <ul style="list-style-type: none"> • New message ID indicates that reports or replies generated as a result of this message, a new <i>msgId</i> is generated for the report or reply message. • Pass Message ID indicates that if a report or reply is generated as a result of this message, then the <i>msgId</i> of this message is copied to the <i>msgId</i> of the report or reply message. • copy Message ID to Correlation ID indicates that if a report or reply is generated as a result of this message, then the <i>msgId</i> of this message is copied to the <i>correlId</i> of the report or reply message. • Pass Correlation ID indicates that if a report or reply is generated as a result of this message, then the <i>correlId</i> of this message is copied to the correlId of the report or reply message.

Related tasks

[“Creating a new service definition” on page 230](#)

The service definition wizard simplifies the process of creating service definitions and is integrated into the IBM MQ Explorer. The service definition wizard is deprecated in IBM MQ 8.0

[“Adding a service definition repository” on page 229](#)

Use this information to create a new service definition repository.

[“Configuring queue managers and objects” on page 51](#)

You can configure many of the attributes of queue managers and their objects from IBM MQ Explorer using the properties dialogs.

IBM MQ subscription properties

You can set attributes for all types of subscriptions. Some of the attributes do not apply to all types of subscriptions, some attributes are specific to z/OS subscriptions.

The following tables list all the attributes that you can set:

- [General](#)
- [Extended](#)
- [Statistics](#)

For each attribute, there is a brief description of when you might need to configure the attribute. The tables also give the equivalent MQSC parameters. For more information about MQSC commands, see [Script \(MQSC\) Commands](#) in the IBM MQ online product documentation.

General page

The following table lists the attributes you can set on the **General** page of the **Subscription properties** dialog.

Attribute	Meaning	MQSC parameter
Subscription name	Read-only. You cannot change the name of the subscription after it has been created.	SUBNAME
Topic name	The name of the topic object used by the subscription. The topic name provides an optional topic root. A maximum of 48 characters are allowed.	TOPICOBJ
Topic string	<p>This specifies a fully qualified topic name or wildcard topic set for the subscription.</p> <p>The slash (/) 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.</p>	TOPICSTR
Wildcard usage	<p>The scheme is used when interpreting any wildcard characters contained in the Topic string. The two values are:</p> <p>TOPIC: Wildcard characters represent portions of the topic hierarchy.</p> <p>CHAR: Wildcard characters represent portions of strings.</p>	WSHEMA

Attribute	Meaning	MQSC parameter
Scope	<p>The scope determines whether this subscription is forwarded to other queue managers, so that the subscriber receives messages published at those other queue managers. The two values are:</p> <p>ALL The subscription is forwarded to all queue managers directly connected through a publish/subscribe collective or hierarchy.</p> <p>QMGR The subscription forwards messages published on the topic only within this queue manager.</p> <p>Note: Individual subscribers can only restrict SUBSCOPE. If the parameter is set to ALL at topic level, then an individual subscriber can restrict it to QMGR for this subscription. However, if the parameter is set to QMGR at topic level, then setting an individual subscriber to ALL has no effect.</p>	SUBSCOPE
Destination class	<p>The Destination class specifies whether the destination used by the subscription is a managed destination. The two values are:</p> <p>MANAGED: The destination is managed.</p> <p>PROVIDED: The destination is a queue. This is the default value.</p>	DESTCLAS
Destination queue manager	The destination queue manager for messages published to the subscription.	DESTQMGR
Destination name	Specifies the name of the alias, local, remote, or cluster queue to which messages for this subscription are put.	DEST

Attribute	Meaning	MQSC parameter
Properties	<p>Properties determines how pub/sub related message properties are added to messages sent to the subscription. The available options are:</p> <p>Compatibility: Publish / subscribe properties are added to the message to maintain compatibility with IBM WebSphere MQ 6.0 Publish / Subscribe.</p> <p>Message properties: Publish / subscribe properties are added as message properties.</p> <p>None: Publish / subscribe properties are not added to the message.</p> <p>RFH2: Publish / subscribe properties are added to the message within an RFH 2 header.</p>	PSPROP
User data	The value of User data can be optionally passed as a message property in a message sent to the subscription.	USERDATA
Selector	The Selector is an SQL92 string that is applied to messages published on the named topic to select whether they are eligible for the subscription.	SELECTOR
Selector type	The type of SelectionString that has been specified. This display attribute is calculated and is not associated with an object. The selector type will be filterable (for example, with a WHERE clause) to allow an administrator to display only internal or only external selectors.	SELTYPE

Extended page

The following table lists the attributes you can set on the **Extended** page of the **Subscription properties** dialog.

Attribute	Meaning	MQSC parameter
Variable user ID	<p>Specifies whether users other than the creator of the subscription, can connect to it and take ownership of the subscription. The two values are:</p> <p>ANY: Other users can connect to the subscription if the other user has the correct topic authority checks and destination authority checks. This is the default value.</p> <p>FIXED: Other users cannot connect to the subscription.</p>	VARUSER
User	Specifies the user profile that owns this subscription.	SUBUSER
Application identity data	<p>The value of Application identity data will be used for messages sent to the subscription. If Application identity data is not specified, then an empty default value is used.</p>	PUBAPPID
Accounting token	<p>The value of Accounting token will be used for messages sent to the subscription. If Accounting token is not specified, then the default value MQACT_NONE is used.</p>	PUBACCT
Publish priority	<p>The Publish priority determines the manner in which pub/sub related message properties are added to the messages sent to the subscription. The options available are:</p> <p>As published which means the priority of the message sent to this subscription and is taken from that supplied in the published message.</p> <p>As queue defined which means the priority of the message sent to this subscription and is taken from the default priority of the queue defined as the destination.</p> <p>Priority-value which enables you to specify a priority ranging 0 - 9.</p>	PUBPRTY

Attribute	Meaning	MQSC parameter
Subscription ID	The value of Subscription ID is assigned by the queue manager as an all time unique identifier for the subscription. This identifier can be used as an alternative to SUBNAME as the target for DISPLAY , ALTER , and DELETE MQSC commands when it is not possible to provide the SUBNAME due to its format, or if no SUBNAME was provided for an application created subscription.	SUBID
Expiry	The time to live of the subscription from the creation date and time. Expiry is measured in tenths of a second. The two values are: Unlimited which means the subscription will never expire, or the user can type their own value which is measured in tenths of a second. Zero is the default value.	EXPIRY
Request only	Request only indicates whether the subscriber will poll for updates via MQSUBPRQ API . The two values are: All which means that all publications are delivered to the subscription. this is the default value. On request which means that publications are only delivered to the subscription in response to MQSUBPRQ API .	REQONLY
Subscription level	This is the level associated with the subscription. Publications will only be delivered to this subscription if it is in the set of subscriptions with the highest SubLevel value less than or equal to the PubLevel used at publication time. The value must be in the range 0 - 9. Zero is the lowest level.	SUBLEVEL

Statistics page

The following table lists the attributes on the **Statistics** page of the **Subscription properties** dialog. The **Statistics** page displays information about the history of the subscription. The information displayed on the **Statistics** page is read-only and cannot be altered by the user.

Attribute	Meaning	MQSC parameter
Creation Date	Read-only. This is the date on which the subscription was created.	CRDATE

Attribute	Meaning	MQSC parameter
Creation Time	Read-only. this is the time at which the subscription was created.	CRTIME
Alteration Date	Read-only. This is the date on which the subscription's attributes were last altered.	ALTDATE
Alteration Time	Read-only. This is the time at which the subscription's attributes were last altered.	ALTTIME

Related concepts

[“Publishers and subscribers” on page 117](#)

Publishers and subscribers are applications that send and receive messages (publications) using the publish/subscribe method of messaging. Publishers and subscribers are decoupled from one another so that publishers do not know the destination of the information that they send, and subscribers do not know the source of the information that they receive.

Related tasks

[“Configuring queue managers and objects” on page 51](#)

You can configure many of the attributes of queue managers and their objects from IBM MQ Explorer using the properties dialogs.

[“Forcing changes to queue properties” on page 53](#)

If the changes that you are making to the attributes of a queue affect the operation of a queue manager or another program, you might be asked to confirm whether you want to force the changes to the queue attributes.

Related reference

[“Strings in property dialogs” on page 655](#)

To include certain characters in a string, you must punctuate the string in a particular way.

Process definition properties

You can set attributes for process definitions. Some attributes do not apply to all types of process definitions. Some of the attributes are specific to z/OS process definitions.

The following tables list all the attributes that you can set for process definitions:

- [General](#)
- [Statistics](#)

For each attribute, there is a brief description of when you might need to configure the attribute. The tables also give the equivalent MQSC parameter for the ALTER PROCESS and DISPLAY PROCESS commands. For more information, see [ALTER PROCESS](#) and [DISPLAY PROCESS](#) in the IBM MQonline product documentation.

General page

The following table lists the attributes that you can set on the **General** page of the Process Definition properties dialog.

Attribute	Meaning	MQSC parameter
Process name	Read-only. You cannot change the name of the process definition after it has been created.	PROCESS

Attribute	Meaning	MQSC parameter
Description	Type a meaningful description of the purpose of the process definition. See Entering strings in MQ Explorer .	DESCR
Application type	<p>Select the type of application that starts when the initiation queue receives the trigger message.</p> <p>System-defined applications are in the range zero through 65 535. For user-defined applications, type in the range 65 536 through 999 999 999.</p> <p>Only specify application types (other than user-defined types) that are supported on the platform at which the command is executed:</p> <ul style="list-style-type: none"> • z/OS supports CICS (default), DOS, IMS, MVS™, OS2, UNIX, Windows, Windows NT, and DEF. • OS/400® supports OS/400 (default), CICS, and DEF. • Tandem NSK supports NSK. • UNIX supports UNIX (default), OS2, DOS, Windows, CICS, and DEF. • Windows NT supports Windows NT (default), DOS, Windows, OS2, UNIX, CICS, and DEF. 	APPLTYPE
Application ID	Type the name of the application to be started. Usually, this is the fully-qualified file name of the executable object. The maximum length is 256 characters. For a CICS application, type the CICS transaction ID; for an IMS application, type the IMS transaction ID.	APPLICID
Environment data	Type the environment information that is relevant to the application being started. The maximum length is 128 characters.	ENVRDATA
User data	Type the user information that is relevant to the application being started. The maximum length is 128 characters.	USERDATA

Attribute	Meaning	MQSC parameter
QSG disposition	Read-only. This is the queue sharing group disposition of the process definition. You cannot change the disposition of a process definition after it has been created. Queue manager means that the object definition is available only to the queue manager that hosts it; Group means that the object definition is stored on the shared repository and each queue manager in the queue-sharing group has a copy of the definition; Copy means that the object definition is the queue manager's copy of a definition in the shared repository.	QSGDISP

Statistics page

The following table lists the attributes that you can set on the **Statistics** page of the Process Definitions properties dialog. The **Statistics** page displays information about the history of the process definitions. You cannot edit any of these attributes.

Attribute	Meaning	MQSC parameter
Alteration date	Read-only. This is the date on which the process definition's attributes were last altered.	ALTDATE
Alteration time	Read-only. This is the time at which the process definition's attributes were last altered.	ALTTIME

Related tasks

[“Configuring queue managers and objects” on page 51](#)

You can configure many of the attributes of queue managers and their objects from IBM MQ Explorer using the properties dialogs.

Related reference

[“Strings in property dialogs” on page 655](#)

To include certain characters in a string, you must punctuate the string in a particular way.

Namelist properties

You can set attributes for namelists. Some of the attributes are specific to z/OS namelists.

The following tables list the attributes that you can set:

- [General](#)
- [Statistics](#)

For each attribute, there is a brief description of when you might need to configure the attribute. The tables also give the equivalent MQSC parameter for the ALTER NAMELIST and DISPLAY NAMELIST commands. For more information about MQSC commands, see [Script \(MQSC\) Commands](#) in the IBM MQ online product documentation.

General page

The following table lists the attributes that you can set on the **General** page of the Namelist properties dialog.

Attribute	Meaning	MQSC parameter
Namelist name	Read-only. You cannot change the name of the namelist after it has been created.	NAMELIST
Description	Type a meaningful description of the purpose of the namelist. See Entering strings in MQ Explorer .	DESCR
Names	Type the list of names of objects that are associated with the namelist. The objects must be defined on the local queue manager. See Entering strings in MQ Explorer .	NAMES
Name count	Read-only. This is the number of names in the namelist.	NAMCOUNT
QSG disposition	Read-only. This is the queue sharing group disposition of the namelist. You cannot change the disposition of a namelist after it has been created. Queue manager means that the object definition is available only to the queue manager that hosts it; Group means that the object definition is stored on the shared repository and each queue manager in the queue sharing group has a copy of the definition; Copy means that the object definition is the queue manager's copy of a definition in the shared repository.	QSGDISP

Statistics page

The following table lists the attributes that you can set on the **Statistics** page of the Namelist properties dialog. The **Statistics** page displays information about the history of the namelist. You cannot edit any of these attributes.

Attribute	Meaning	MQSC parameter
Alteration date	Read-only. This is the date on which the namelist's attributes were last altered.	ALTDATE
Alteration time	Read-only. This is the time at which the namelist's attributes were last altered.	ALTTIME

Related tasks

[“Configuring queue managers and objects” on page 51](#)

You can configure many of the attributes of queue managers and their objects from IBM MQ Explorer using the properties dialogs.

Related reference

[“Strings in property dialogs” on page 655](#)

To include certain characters in a string, you must punctuate the string in a particular way.

Authentication information properties

You can set attributes for all types of authentication information objects. Some of the attributes do not apply to all types of authentication information objects, and some attributes are specific to z/OS authentication information objects.

The following tables list the attributes that you can set:

- [General](#)
- [LDAP](#)
- [OCSP](#)
- [LDAP User Repository](#)
- [LDAP Authorization](#)
- [User ID + Password](#)
- [Statistics](#)

For each attribute, there is a brief description of when you might need to configure it. The tables also give the equivalent MQSC parameter for the ALTER AUTHINFO and DISPLAY AUTHINFO commands. For more information about MQSC commands, see [Script \(MQSC\) Commands](#) in the IBM MQ online product documentation.

General page

The following table lists the attributes that you can set on the **General** page of the Authentication Information properties dialog.

Attribute	Meaning	MQSC parameter
Authinfo name	Read-only. You cannot change the name of an authentication information object after it has been created.	AUTHINFO
Authinfo type	Read-only. You cannot change the type of an authentication information object after it has been created.	AUTHTYPE
Description	Type a meaningful description of the purpose of the authentication information object. See Entering strings in MQ Explorer .	DESCR

Attribute	Meaning	MQSC parameter
QSG disposition	Read-only. The queue sharing group disposition of the authentication information object. You cannot change the disposition of an authentication information object after it has been created. Queue manager means that the object definition is available only to the queue manager that hosts it; Group means that the object definition is stored on the shared repository and each queue manager in the queue sharing group has a copy of the definition; Copy means that the object definition is the queue manager's copy of a definition in the shared repository.	QSGDISP

LDAP page

The following table lists the attributes that you can set on the **LDAP** page of the CRL LDAP or IDPW LDAP Authentication Information properties dialog. The **LDAP** page displays the name and authentication information for the LDAP server.

Attribute	Meaning	MQSC parameter
LDAP Server Name	Type the host name, IPv4 dotted decimal address, or IPv6 hexadecimal notation of the host on which the LDAP server is running, with an optional port number. If you specify the connection name as an IPv6 address, only systems that are running IBM WebSphere MQ 6.0 authentication information objects with an IPv6 stack are able to resolve this address. If the authentication information object is part of the queue manager's CRL namelist, ensure that any clients that are using the client channel table that is generated by the queue manager are capable of resolving the connection name. On z/OS, to use a connection name that resolves to an IPv6 network address, the level of z/OS must support IPv6 for connecting to an LDAP server.	CONNNAME

Attribute	Meaning	MQSC parameter
User ID	Type the Distinguished Name of the user who is accessing the LDAP server, with the following limitations: <ul style="list-style-type: none"> On IBM i, UNIX, and Windows, the maximum length is 1024 characters. On z/OS, the maximum length is 256 characters. If you use asterisks (*) in the user name, they are treated as literal characters, and not as wildcards, because the LDAP user ID is a specific name and not a string used for matching. 	LDAPUSER
Password	Type the password that is associated with the Distinguished Name of the user who is accessing the LDAP server. The maximum length is 32 characters.	LDAPPWD

OCSP page

The following table lists the attributes that you can set on the **OCSP** page of the OCSP Authentication Information properties dialog.

Attribute	Meaning	MQSC parameter
OCSP responder URL	The URL at which the OCSP responder can be contacted. This attribute takes priority over a URL in an AuthorityInfoAccess (AIA) certificate extension.	OCSPURL

LDAP User Repository page


The following table lists the attributes that you can set on the **LDAP User Repository** page of the IDPW LDAP Authentication Information properties dialog.

Attribute	Meaning	MQSC parameter
Equivalent short user	A field in the LDAP user record to be used as a short user name for this connection.	SHORTUSR
User ID base DN	The base DN used to locate user records in an LDAP server.	BASEDNU
Use secure communication	Whether the connection to the LDAP server will be made using TLS.	SECCOMM
User Object Class	The LDAP object class used for user records in the LDAP repository.	CLASSUSR

Attribute	Meaning	MQSC parameter
Qualifying user field	A qualification to allow user IDs provided by applications to be identified as a field in the LDAP user record.	USRFIELD

LDAP Authorization

The following table lists the attributes that you can set on the **LDAP Authorization** page of the IDPW LDAP Authentication Information properties dialog.

Attribute	Meaning	MQSC parameter
Authorization method	<p>Whether authorization is done using user IDs and groups from the Operating System or from LDAP. The possible values are:</p> <p>Operating System. Authorization is done using user IDs and groups from the Operating System.</p> <p>Search group. Authorization is done using user IDs and groups from LDAP. The group entry in the LDAP repository contains an attribute listing the Distinguished Name of all the users who belong to the group.</p> <p>Search user. Authorization is done using user IDs and groups from LDAP. The user entry in the LDAP repository contains an attribute listing all the Distinguished Name of the groups to which the user belongs.</p> <p> Search group short name. Authorization is done using user IDs and groups from LDAP. The group entry in the LDAP repository contains an attribute listing the short user name of all the users who belong to the group.</p>	AUTHORMD
Allow nested groups	<p>Whether nested groups are allowed. The possible values are:</p> <p>No. Nested groups are not allowed.</p> <p>Yes. Nested groups are allowed. The group list is searched recursively to enumerate all groups a user belongs to.</p>	NESTGRP
Group base DN	The base DN used to locate group records in an LDAP server.	BASEDNG

Attribute	Meaning	MQSC parameter
Group object class	The LDAP object class used for group records in the LDAP repository.	CLASSGRP
Qualifying group field	A qualification to allow group to be identified as a field in the LDAP group record.	GRPFIELD
Group membership field	Name of the attribute used within an LDAP user or group record to determine group membership.	FINDGRP

User ID + Password page

The following table lists the attributes that you can set on the **User ID + Password** page of the IDPW OS or IDPW LDAP Authentication Information properties dialog.

Attribute	Meaning	MQSC parameter
Check locally bound connections	<p>Whether connections made by using local bindings, connections must supply a user ID and password for validation. The possible values are:</p> <p>None. No user ID and password are required.</p> <p>Optional. No user ID and password are required but if provided, they are checked.</p> <p>Required for administrators. User ID and password are required for privileged users.</p> <p>Required for all. User ID and password are required for all users.</p> <p>Setting CHCKLOCL to Required for administrators or Required for all results in the inability to locally administer the queue manager by way of the runmqsc commands unless you specify the -u UserID parameter on the runmqsc command line. If you do not specify this parameter, you see error message AMQ8135: Not authorized. Similarly, when you run IBM MQ Explorer on your local system, you might see error AMQ4036: Access not permitted when you are attempting to connect to a queue manager.</p> <p>To specify a user name and password, right-click the local queue manager object, and select Connection Details > Properties from the menu. In the UserID section, enter the user name and password, and then click OK.</p>	CHCKLOCL

Attribute	Meaning	MQSC parameter
Check client connections	<p>Whether connections made using client connections must supply a user ID and password for validation. The possible values are:</p> <p>None. No user ID and password are required.</p> <p>Optional. No user ID and password are required but if provided, they will be checked.</p> <p>Required for administrators. User ID and password are required for privileged users.</p> <p>Required for all. User ID and password are required for all users.</p>	CHCKCLNT
Adopt the authenticated user	<p>Whether to adopt the user ID that was provided with a password as the context for this connection. The possible values are:</p> <p>Yes. The validated user ID will be adopted as the context for this connection. If the user ID presented is an LDAP user ID, and authorization checks are done using operating system user IDs, the SHORTUSR associated with the user entry in LDAP will be adopted as the credentials for authorization checks to be done against.</p> <p>No. The validated user ID will not be adopted as the context for this connection.</p>	ADOPTCTX
Authentication failure delay	<p>This attribute specifies how long to delay before returning the failure return code to the application, for example, if no response is received by a mqmconnx request. This is the length of time in seconds, which can be 0 - 60. A value of zero means that no delay is added.</p>	FAILDLAY

Statistics page

The following table lists the attributes that you can set on the **Statistics** page of the Authentication Information properties dialog. The **Statistics** page displays information about the history of the authentication information object. You cannot edit the values of any of these attributes.

Attribute	Meaning	MQSC parameter
Alteration date	Read-only. This is the date on which the authentication information object attributes were last altered.	ALTDATE
Alteration time	Read-only. This is the time at which the authentication information object attributes were last altered.	ALTTIME

Related tasks

[“Configuring queue managers and objects” on page 51](#)

You can configure many of the attributes of queue managers and their objects from IBM MQ Explorer using the properties dialogs.

Related reference

[“Strings in property dialogs” on page 655](#)

To include certain characters in a string, you must punctuate the string in a particular way.

Channel authentication record properties

You can set attributes for channel authentication record objects.

The following tables list the attributes that you can set:

- [General](#)
- [Address](#)
- [Block address](#)
- [Block user](#)
- [Queue manager](#)
- [SSL peer](#)
- [Client user](#)
- [Extended](#)
- [Statistics](#)

For each attribute, there is a brief description of when you might need to configure it. The tables also give the equivalent MQSC parameter for the SET CHLAUTH and DISPLAY CHLAUTH commands. For more information about MQSC commands, see [Script \(MQSC\) Commands](#) in the IBM MQ online product documentation.

General page

The following table lists the attributes that you can set on the **General** page of the **Channel Authentication Records** properties dialog.

Attribute	Meaning	MQSC parameter
Channel profile	Channel profile name. See SET CHLAUTH in the IBM MQ online product documentation.	PROFILE
Type	Can be Address Map, Block Address List, Block User List, Queue Manager Map, SSL Peer Map or User Map. See SET CHLAUTH in the IBM MQ online product documentation.	TYPE

Attribute	Meaning	MQSC parameter
Description	Type a meaningful description of the purpose of the channel authentication record. See Entering strings in MQ Explorer .	DESCR

Address page

The following table lists the attributes that you can set on the **Address** page of the **Channel Authentication Records** properties dialog.

Note:

This parameter is valid with the property TYPE(ADDRESSMAP), TYPE(QMGRMAP), TYPE(SSLPEERMAP) and TYPE(USERMAP).

Attribute	Meaning	MQSC parameter
Address	Specifies the filter to be used to compare with the IP address of the partner queue manager or client at the other end of the channel. For SET command this parameter is mandatory with TYPE(ADDRESSMAP). See SET CHLAUTH in the IBM MQ online product documentation.	ADDRESS

Block address page

The following table lists the attributes that you can set on the **Block address** page of the **Channel Authentication Records** properties dialog.

Note:

This parameter is only valid with the property TYPE(BLOCKADDR).

Attribute	Meaning	MQSC parameter
Address list	A list of IP address patterns which are blocked from connecting to this queue manager using any channel. See SET CHLAUTH in the IBM MQ online product documentation.	ADDRLIST

Block user page

The following table lists the attributes that you can set on the **Block user** page of the **Channel Authentication Records** properties dialog.

Note:

This parameter is only valid with the property TYPE(BLOCKUSER).

Attribute	Meaning	MQSC parameter
User list	A list of user IDs that are blocked from use of this channel or set of channels. See SET CHLAUTH in the IBM MQ online product documentation.	USERLIST

Queue manager page

The following table lists the attributes that you can set on the **Queue manager** page of the **Channel Authentication Records** properties dialog.

Note:

This parameter is only valid with the property TYPE(QMGRMAP).

Attribute	Meaning	MQSC parameter
Remote queue manager	Specifies the remote partner queue manager name pattern. See SET CHLAUTH in the IBM MQ online product documentation.	QMNAME

SSL peer page

The following table lists the attributes that you can set on the **SSL peer** page of the **Channel Authentication Records** properties dialog.

Note:

This parameter is only valid with the property TYPE(SSLPEERMAP).

Attribute	Meaning	MQSC parameter
Peer name	The value of the Distinguished Name on the certificate from the peer queue manager or client at the other end of the IBM MQ channel. When the channel starts, the value of this attribute is compared with the Distinguished Name of the certificate. See SET CHLAUTH in the IBM MQ online product documentation.	SSLPEER
SSL/TLS issuer's Distinguished Name	If this optional parameter is specified, it only allows connections from partner queue managers for which the certificate was issued by a Certificate Authority with a matching Distinguished Name. See SET CHLAUTH in the IBM MQ online product documentation.	SSLCERTI

Client user page

The following table lists the attributes that you can set on the **Client user** page of the **Channel Authentication Records** properties dialog.

Note:

This parameter is only valid with the property TYPE(USERMAP).

Attribute	Meaning	MQSC parameter
Client user ID	Specifies the client asserted user ID. See SET CHLAUTH in the IBM MQ online product documentation.	CLNTUSER

Extended page

The following table lists the attributes that you can set on the **Extended** page of the **Channel Authentication Records** properties dialog. For more information on attributes on this page see [SET CHLAUTH](#) in the IBM MQ online product documentation.

Attribute	Meaning	MQSC parameter
User source	Source of the user ID to be used for MCAUSER at run time. Possible values are Channel, Map and No access.	USERSRC
MCA user ID	Message channel user ID to be used when the inbound connection matches the SSL DN, IP address, client asserted user ID or remote queue manager name supplied. This attribute is enabled only when User source selected is Map.	MCAUSER
Warning	Indicates whether this record should operate in warning mode. Possible values are Yes or No.	WARN
Check client connection	Specifies whether the connection that matches this rule and is being allowed in with USERSRC(CHANNEL) or USERSRC(MAP), must also specify a valid user ID and password.	CHCKCLNT
Custom	This attribute is reserved for the configuration of new features before separate attributes have been introduced.	CUSTOM

Statistics page

The **Statistics** page of the **Channel Authentication Records** properties dialog displays read-only information showing when the properties of the channel authentication record were last changed. You cannot edit the values of these attributes. See [DISPLAY CHLAUTH](#) in the IBM MQ online product documentation.

Attribute	Meaning	MQSC parameter
Alteration date	Read-only. This is the date on which the authentication information object attributes were last altered.	ALTDATE

Attribute	Meaning	MQSC parameter
Alteration time	Read-only. This is the time at which the authentication information object attributes were last altered.	ALTTIME

Related reference

[“Strings in property dialogs” on page 655](#)

To include certain characters in a string, you must punctuate the string in a particular way.

Related information

[Channel authentication records](#)

[SET AUTHREC](#)

[Message channel agent user identifier \(MCAUSER\)](#)

Multicast Communication Information object properties

You can set attributes for Multicast communication information objects.

The following tables list the attributes that you can set:

- [General](#)
- [Statistics](#)







For each attribute, there is a brief description of when you might need to configure the attribute. The tables also give the equivalent MQSC parameter for the ALTER COMMINFO and DISPLAY COMMINFO commands. For more information about MQSC commands, see [The MQSC Commands](#) in the IBM MQ online product documentation.

General page

The following table lists the attributes that you can set on the **General** page of the communication information object properties dialog.

Attribute	Meaning	MQSC parameter
Communication Information	Read only after it has been defined. The name of the communications information object. The name must not be the same as any other communications information object name currently defined on this queue manager. For more information, see Rules for naming IBM MQ objects in the IBM MQ online product documentation.	COMMINFO
Type	Read only after it has been defined. The type of the communications information object. The only type supported is MULTICAST .	TYPE

Attribute	Meaning	MQSC parameter
Description	<p>Plain text comment. The description provides descriptive information about the communication information object when an operator issues the DISPLAY COMMINFO command. For more information, see DISPLAY COMMINFO in the IBM MQ online product documentation.</p> <p>The description must contain only displayable characters. The maximum length is 64 characters. In a DBCS installation, it can contain DBCS characters (subject to a maximum length of 64 bytes).</p> <p>Note: If characters are used that are not in the coded character set identifier (CCSID) for this queue manager, they might be translated incorrectly if the information is sent to another queue manager.</p>	DESCR
Group address	<p>The group IP address or DNS name. It is the administrator's responsibility to manage the group addresses.</p> <p>It is possible for all multicast clients to use the same group address for every topic; only the messages that match outstanding subscriptions on the client are delivered.</p> <p>Using the same group address can be inefficient because every client must examine and process every multicast packet in the network. It is more efficient to allocate different IP group addresses to different topics or sets of topics, but this requires careful management, especially if other non-MQ multicast applications are in use on the network. The default value is 239.0.0.0.</p>	GRPADDR
Port	<p>The port number to transmit on. The default port number is 1414</p>	PORT

Attribute	Meaning	MQSC parameter
Message history	<p>The maximum message history is the amount of message history that is kept by the system to handle retransmissions in the case of NACKs (negative acknowledgments).</p> <p>A value of 0 gives the least level of reliability. The default value is 100 messages.</p>	MSGHIST
Coded character set ID	<p>The coded character set identifier that messages are transmitted on. Specify a value in the range 1 through 65535, or set it to As published which is the default value.</p> <p>The CCSID must specify a value that is defined for use on your platform, and use a character set that is appropriate to the platform. If you use this parameter to change the CCSID, applications that are running when the change is applied continue to use the original CCSID. Because of this, you must stop and restart all running applications before you continue. This includes the command server and channel programs.</p> <p>To do this, stop and restart the queue manager after making the change. This parameter is valid only on the following platforms:</p> <ul style="list-style-type: none"> •  AIX •  HP-UX •  IBM i •  Linux •  Solaris •  Windows <p>For details of the supported CCSIDs for each platform, see Code page conversion in the IBM MQ online product documentation.</p>	CCSID

Attribute	Meaning	MQSC parameter
Encoding	<p>The encoding that the messages are transmitted in.</p> <ul style="list-style-type: none"> • As published. This is the default value. • Reversed • Normal • S390 • TNS • encoding 	ENCODING
New subscriber history	<p>The new subscriber history controls whether a subscriber joining a publication stream receives as much data as is currently available, or receives only publications made from the time of the subscription.</p> <ul style="list-style-type: none"> • None. A value of None causes the transmitter to transmit only publication made from the time of the subscription. This is the default value. • ALL. A value of ALL causes the transmitter to retransmit as much history of the topic as is known. In some circumstances, this can give a similar behavior to retained publications. 	NSUBHIST
Monitor interval (milliseconds)	<p>How frequently, in seconds, that monitoring information is updated. If events messages are enabled, this parameter also controls how frequently event messages about the status of the Multicast handles created using this COMMINFO object are generated.</p> <p>A value of 0 means that there is no monitoring. The default value is 60.</p>	MONINT

Attribute	Meaning	MQSC parameter
Communication events	<p>Controls whether event messages are generated for Multicast handles that are created using this COMMINFO object. Events will only be generated if they are enabled using the MONINT parameter. The three possible values are:</p> <ul style="list-style-type: none"> • Disabled. Publications from applications not using Multicast are not bridged to applications that do use Multicast. This is the default value. • Enabled. Publications from applications not using Multicast are bridged to applications that do use Multicast. • Exception. Event messages are written if the message reliability is below the reliability threshold. The reliability threshold is set to 90. 	COMMEV
Multicast bridge	<p>Controls whether publications from applications not using Multicast are bridged to applications using Multicast. Bridging does not apply to topics that are marked as MCAST(ONLY). As these topics can only be Multicast traffic, it is not applicable to bridge to the queue publish/subscribe domain. The two possible values are:</p> <ul style="list-style-type: none"> • Disabled. Publications from applications not using Multicast are not bridged to applications that do use Multicast. This is the default for i5/OS. • Enabled. Publications from applications not using Multicast are bridged to applications that do use Multicast. This is the default for platforms other than i5/OS. 	BRIDGE
Multicast heartbeat interval (milliseconds)	<p>The heartbeat interval is measured in milliseconds, and specifies the frequency at which the transmitter notifies any receivers that there is no further data available. The default value is 2000 milliseconds.</p>	MCHBINT

Attribute	Meaning	MQSC parameter
Multicast property control	<p>The multicast properties control how many of the MQMD properties and user properties flow with the message.</p> <ul style="list-style-type: none"> • All. All user properties and all the fields of the MQMD are transported. This is the default value. • Reply. Only user properties, and MQMD fields that deal with replying to the messages, are transmitted. These properties are: <ul style="list-style-type: none"> – MsgType – MessageId – CorrelId – ReplyToQ – ReplyToQmgr • User. Only the user properties are transmitted. • None. No user properties or MQMD fields are transmitted. • Compatible. This value causes the transmission of the message to be done in a compatible mode to RMM. This allows some inter-operation with the current XMS applications and Broker RMM applications. 	MCPROP

Statistics page

The following table lists the attributes that you can set on the **Statistics** page of the **Communication Information** properties dialog. The **Statistics** page displays information about the history of the communication information object. You cannot edit any of these attributes.

Attribute	Meaning	MQSC parameter
Alteration date	Read-only. This is the date on which the communication information object's attributes were last altered.	ALTDATE
Alteration time	Read-only. This is the time at which the communication information object's attributes were last altered.	ALTTIME

Related tasks

[“Configuring queue managers and objects” on page 51](#)

You can configure many of the attributes of queue managers and their objects from IBM MQ Explorer using the properties dialogs.

Storage class properties

You can set attributes for storage classes. Storage classes are available only on z/OS.

The following table lists the attributes that you can set.

For each attribute, there is a brief description of when you might need to configure the attribute. The tables also give the equivalent MQSC parameter for the ALTER STGCLASS and DISPLAY STGCLASS commands. For more information about MQSC commands, see [Script \(MQSC\) Commands](#) in the IBM MQ online product documentation.

General page

The following table lists the attributes that you can set on the **General** page of the **Storage class properties** dialog.

Attribute	Meaning	MQSC parameter
Storage class name	Read-only. You cannot change the name of the storage class after it has been created.	STGCLASS
Description	Type a meaningful description of the purpose of the storage class. See Entering strings in MQ Explorer .	DESCR
PageSet ID	This is the page set identifier that the storage class is associated with. Type a number, from 00 to 99, two characters long. If this attribute is blank, the value is taken from the default storage class, SYSTEMST.	PSID
QSG disposition	Read-only. The queue sharing group disposition of the storage class. You cannot change the disposition of a storage class after it has been created. Queue manager means that the object definition is available only to the queue manager that hosts it; Group means that the object definition is stored on the shared repository and each queue manager in the queue sharing group has a copy of the definition; Copy means that the object definition is the queue manager's copy of a definition in the shared repository.	QSGDISP

Attribute	Meaning	MQSC parameter
XCF group name	If you are using the IMS bridge, this is the name of the XCF group to which the IMS system belongs. Type a name from 1 to 8 characters long. The first character must be an uppercase letter from A to Z; subsequent characters must be uppercase letters from A to Z, numbers from 0 to 9, or both.	XCFGNAME
XCF member name	If you are using the IMS bridge, this is the XCF member name of the IMS system within the XCF group that is specified in the XCF group name attribute. Type a name from 1 to 16 characters long. The first character must be an uppercase letter from A to Z; subsequent characters must be uppercase letters from A to Z, numbers from 0 to 9, or both.	XCFMNAME
Passticket appl name	This is the application name that is passed to RACF when authenticating the passticket that is specified in the MQIIH header. If you do not specify a value, the validation process uses the z/OS Batch Job Profile Name, which means that RACF validates using a profile in the form of MVSxxxx, where xxxx is the SMFID of the z/OS system on which the queue manager is running.	PASSTKTA
Alteration date	Read-only. This is the date on which the storage class's attributes were last altered.	ALTDATE
Alteration time	Read-only. This is the time at which the storage class's attributes were last altered.	ALTTIME

Related tasks

[“Configuring queue managers and objects” on page 51](#)

You can configure many of the attributes of queue managers and their objects from IBM MQ Explorer using the properties dialogs.

Related reference

[“Strings in property dialogs” on page 655](#)

To include certain characters in a string, you must punctuate the string in a particular way.

Queue sharing group properties

You cannot edit the values of any queue sharing group attributes.

The following tables list all the attributes for queue sharing groups.

For each attribute, there is a brief description of what it means. The equivalent MQSC command is DISPLAY GROUP, which returns information about the queue sharing group to which a queue manager belongs. For more information about MQSC commands, see [Script \(MQSC\) Commands](#) in the IBM MQ online product documentation.

You cannot edit the values of any of these attributes.

Attribute	Meaning
Queue sharing group name	The name of the queue sharing group.
Queue manager name	The name of the queue manager.
Queue manager number	The number, generated internally, of the queue manager in the group.
Db2 name	The name of the Db2 subsystem or group to which the queue manager is to connect.
Queue manager status	The current status of the queue manager. Active means that the queue manager is running; Inactive means that the queue manager is not running, having terminated normally; Failed means that the queue manager is not running, having terminated abnormally; Created means that the queue manager has been defined to the group, but has not yet been started; Unknown means that the status cannot be determined.
Db2 connection status	The current status of the connection to Db2.
Command level	The command level supported by the queue manager.
Queue manager CPF	The command prefix of the queue manager.

Related concepts

[“Queue sharing groups” on page 46](#)

Queue sharing groups exist only on z/OS queue managers. A queue sharing group is a group of queue managers that can access the same shared queues. Each member of the queue sharing group has access to the same set of shared queues.

Coupling facility structure properties

You can set attributes for coupling facility structures. Coupling facility structures are available only on z/OS.

For each attribute, there is a brief description of when you might need to configure the attribute. The tables also give the equivalent MQSC parameter for the ALTER CFSTRUCT and DISPLAY CFSTRUCT commands. For more information about MQSC commands, see [Script \(MQSC\) Commands](#) in the IBM MQ online product documentation.

General page

This table lists the attributes that you can set on the **General** page of the **Coupling facility structure properties** dialog.

Attribute	Meaning	MQSC parameter
Coupling facility name	Read-only. You cannot change the name of the coupling facility structure after it has been created.	CFSTRUCT

Attribute	Meaning	MQSC parameter
Coupling facility description	Type a meaningful description of the purpose of the coupling facility structure. See “Strings in property dialogs” on page 655.	DESCR
Level	Read-only. The functional capability level for this coupling facility application structure.	CFLEVEL
Recovery	Read-only. This attribute specifies whether coupling facility structure recovery is supported for the application structure.	RECOVER
Loss of coupling facility connectivity	<p>Specifies the action taken when the queue manager loses connectivity to the CF structures with. The three options are:</p> <ul style="list-style-type: none"> • As queue manager. The action taken is based on the setting of the queue manager CFCONLOS attribute. • Tolerate. The queue manager tolerates a loss of connectivity to CF structures and does not terminate. • Terminate. The queue manager terminates when connectivity to CF structures is lost. <p>This parameter is only valid from CFLEVEL (5) or later. Setting this attribute for a structure at a CFLEVEL earlier than 5 results in PCF reason code MQRCCF_PARM_CONFLICT being returned.</p>	CFCONLOS

Attribute	Meaning	MQSC parameter
Automatic recovery	<p>Specifies the automatic recovery action when a queue manager detects that the structure has failed. Or when a queue manager loses connectivity to the structure and no systems in the SysPlex have connectivity to the Coupling Facility that the structure is allocated in. The value can be any of the following values:</p> <ul style="list-style-type: none"> • Yes. The structure and associated shared message data sets are automatically recovered. • No. The structure is not automatically recovered. <p>This parameter is valid only from CFLEVEL (5) or later. Setting this parameter for a structure at a CFLEVEL earlier than 5 results in PCF reason code MQRCCF_PARM_CONFLICT being returned.</p>	RECAUTO
Alteration date	Read-only. The date on which the coupling facility structure's attributes were last altered.	ALTDATE
Alteration time	Read-only. The time at which the coupling facility structure's attributes were last altered.	ALTTIME
Status	Read-only. The current status of the coupling facility structure.	STATUS

Message offload page

This table lists the properties that you can set on the **Message offload** page of the Coupling facility structure properties dialog.

Property	Meaning	MQSC parameter
Offload	If required, select where the message data for a shared queue is stored. This can be Db2, SMDS (shared message data set) or None if no offload is required.	OFFLOAD
Offload rule 1 threshold (%)	Edit this value to represent your initial threshold for the used capacity of the coupling facility structure. For example 70% would represent that the offload would be triggered when 70% of the coupling facility structure storage was used.	OFFLD1TH

Property	Meaning	MQSC parameter
Offload rule 1 size	Edit this value to represent the size of messages selected to be offloaded when the capacity threshold specified in the Offload rule 1 is reached. All messages exceeding the size specified will be selected to be offloaded. For example 32k would indicate that all messages exceeding 32k would be offloaded.	OFFLD1SZ
Offload rule 2 threshold (%)	Edit this value to represent your secondary threshold for the used capacity of the coupling facility structure. For example 80% would indicate that the offload would be triggered when 80% of the coupling facility structure storage was used.	OFFLD2TH
Offload rule 2 size	Edit this value to represent the size of messages selected to be offloaded when the capacity threshold specified in the Offload rule 2 is reached. All messages exceeding the size specified will be selected to be offloaded. For example 4k would indicate that all messages exceeding 4k would be offloaded.	OFFLD2SZ
Offload rule 3 threshold (%)	Edit this value to represent your final threshold for the used capacity of the coupling facility structure. For example 90% would represent that the offload would be triggered when 90% of the coupling facility structure storage was used.	OFFLD3TH
Offload rule 3 size	Edit this value to represent the size of messages selected to be offloaded when the capacity threshold specified in the Offload rule 3 is reached. All messages exceeding the size specified will be selected to be offloaded. For example 0k would indicate that all remaining messages would be offloaded.	OFFLD3SZ
Generic data set name	Edit this value to provide the generic data set name to be used for the group of shared message data sets associated with this structure.	DSGROUP
Logical block size	Edit this value to provide the logical block size, which is the unit that shared message data set space is allocated to individual queues	DSBLOCK
Number of buffers	Edit this value to provide the number of buffers to be allocated in each queue manager for accessing shared message data sets.	DSBUFS

Property	Meaning	MQSC parameter
Expand data set	Edit this parameter to control whether the queue manager should expand a shared message data set when it becomes nearly full, and further blocks are required in the data set.	EXPAND

Related tasks

[“Configuring queue managers and objects” on page 51](#)

You can configure many of the attributes of queue managers and their objects from IBM MQ Explorer using the properties dialogs.

Related reference

[“Strings in property dialogs” on page 655](#)

To include certain characters in a string, you must punctuate the string in a particular way.

Cluster queue manager properties

The **Cluster Queue Manager properties** dialog shows the attributes of the cluster-sender and cluster-receiver channels on the selected cluster queue manager. You cannot edit any of the attributes in the Cluster Queue Manager properties dialog.

Cluster queue managers are queue managers that are members of a cluster. The term *cluster queue manager* is also used to refer to the records that each queue manager in a cluster maintains about other queue managers and objects in the cluster, in particular, the cluster-sender and cluster-receiver channels.

The **Cluster Queue Manager properties** dialog shows the attributes of the cluster-sender and cluster-receiver channels on the selected cluster queue manager. The following tables list the attributes that are displayed in the properties dialog. These attributes vary from the attributes shown for the same channels in the Channel properties dialog (see [Channel properties](#)).

- [General](#)
- [Extended](#)
- [MCA](#)
- [Exits](#)
- [LU6.2](#)
- [Retry](#)
- [Message retry](#)
- [Cluster](#)
- [SSL](#)
- [Statistics](#)

You cannot edit any of the attributes in the Cluster Queue Manager properties dialog.

The tables give the equivalent MQSC parameter for the DISPLAY CLUSQMGR command. For more information about MQSC commands, see [Script \(MQSC\) Commands](#) in the IBM MQ online product documentation.

General page

The following table lists the attributes on the **General** page of the Cluster Queue Manager properties dialog.

Attribute	Meaning	MQSC parameter
Channel name	The name of the channel definition.	CHANNEL

Attribute	Meaning	MQSC parameter
Channel status	This is the current status of the channel.	STATUS
Connection name	For cluster-sender channels, this is the name of the computer that hosts the target queue manager. For cluster-receiver channels, this is the name of the computer that hosts the local queue manager. The format of the connection name depends on the transmission protocol that is selected.	CONNNAME
Description	A description of the cluster channel.	DESCR
Local communication address	If the channel uses TCP/IP and the channel uses a particular IP address, port, or port range for outbound communications, the local communications address for the channel is specified. The channel binds to the address locally. The format used is <i>ipaddress(low-port, high-port)</i> , where <i>ipaddress</i> is the IP address specified in IPv4 dotted decimal, IPv6 hexadecimal, or alphanumeric host name format. For example, 192.0.2.1 specifies the IPv4 address with any port; 192.0.2.1(1000) specifies the IPv4 address and a specific port; 192.0.2.1(1000,2000) specifies the IPv4 address and a range of ports; (1000) specifies a port only.	LOCLADDR
Suspend	This shows whether the queue manager is suspended from the cluster or not (as a result of the SUSPEND QMGR command). Yes means that the queue manager is suspended.	SUSPEND
Transmission protocol	The transport type that the channel uses.	TRPTYPE

Attribute	Meaning	MQSC parameter
Version	<p>The version of the IBM MQ installation that the cluster queue manager is associated with. The version has the following format:</p> <p>VVRRMMFF</p> <p>VV: Version</p> <p>RR: Release</p> <p>MM: Maintenance level</p> <p>FF: Fix level</p>	VERSION

Extended page

The following table lists the attributes on the **Extended** page of the Cluster Queue Manager properties dialog.

Attribute	Meaning	MQSC parameter
Maximum message length	The maximum length of a message, in bytes, that can be transmitted on the channel.	MAXMSGL
Heartbeat interval	The length of the heartbeat interval in seconds. A value of zero means that no heartbeat exchange takes place. The value that is used is the larger of the values specified at the sending side and the receiving end of the channel. The heartbeat interval is the time, in seconds, between heartbeat flows passed from the sending MCA when there are no messages on the transmission queue. The heartbeat exchange gives the receiving MCA the opportunity to quiesce the channel.	HBINT
Sequence number wrap	The sequence number is the count of messages that are sent through the channel. The sequence number increments each time a message is sent through the channel. This attribute shows the highest message sequence number that can be reached before it restarts at 1. The two ends of the channel must have the same sequence number wrap value when the channel starts; otherwise you get an error.	SEQWRAP

Attribute	Meaning	MQSC parameter
Non-persistent message speed	This shows whether non-persistent messages are sent as part of a transaction. Fast means that non-persistent messages are not sent as part of a transaction and so become available for retrieval far more quickly than if they are part of a transaction; Normal means that non-persistent messages are sent as part of a transaction, which reduces the risk of losing the messages if the channel stops while the messages are in transit.	NPMSPEED
Batch size	The maximum number of messages to be sent before syncpoint is taken. The messages are always transferred individually but are committed or backed out as a batch.	BATCHSZ
Disconnect interval (seconds)	The number of seconds after the batch ends before the channel closes down. On all platforms except for z/OS, a value of 0 means that the channel does not disconnect. On z/OS, a value of 0 means that the channel disconnects immediately.	DISCINT
Data conversion	This shows whether the message is converted before transmission to the format required by the receiving system. Yes means that the message is converted before transmission; No means that the message is converted by the receiving application to the format that is required on the receiving system (this is the typical method).	CONVERT
Batch interval (milliseconds)	The number of milliseconds during which the channel keeps a batch open even if there are no messages on the transmission queue.	BATCHINT

Attribute	Meaning	MQSC parameter
Batch heartbeat interval (milliseconds)	The batch heartbeat interval allows the sending end of the channel to verify that the receiving end of the channel is still active just before the sending end of the channel commits a batch of messages. If the receiving end of the channel is not active, the batch can be backed out rather than becoming in-doubt. By backing out the batch, the messages remain available for processing so that they can, for example, be redirected to another channel. This attribute shows the number of seconds that the sending end of the channel waits for a response from the receiving end of the channel before assuming that the receiving end of the channel is inactive. A value of 0 means that batch heartbeating is not used. For more information, see Reducing the likelihood of a channel being in doubt .	BATCHHB
Put authority	This attribute specifies the type of security processing to be carried out by the Message Channel Agent (MCA) when running an MQPUT command to the target queue or an MQI call. Default means that the default user ID is used; Context means that the alternate user ID from the context information that is associated with the message is used.	PUTAUT

Attribute	Meaning	MQSC parameter
Message compression	<p>This shows the message compression techniques that are supported by the channel definition, in order of preference. The first technique that is supported by the other end of the channel is used. None means that no message compression is performed; RLE means that message data compression is performed using run-length encoding; ZLIBFAST means that message data compression is performed using the zlib compression technique and a fast compression time is preferred; ZLIBHIGH means that message data compression is performed using the zlib compression technique and a high level of compression is preferred; ANY means that any compression technique that is supported by the queue manager can be used. For more information, see Distributed queueing and clusters in the IBM MQ online product documentation.</p>	COMPMSG
Header compression	<p>This shows the header compression techniques that are supported by the channel definition, in order of preference. The first technique that is supported by the other end of the channel is used. None means that no header compression is performed; System means that header compression is performed. For more information, see Distributed queueing and clusters in the IBM MQ online product documentation.</p>	COMPHDR

Attribute	Meaning	MQSC parameter
Property control	<p>This defines what happens to properties of messages that are about to be sent to a V6 or earlier queue manager. The value has to be changed from Compatibility to Force to preserve the V6 behavior of propagating the RFH2 to the caller. The possible values are:</p> <p>All means that all properties of the message are included with the message when it is sent to the remote queue manager. The properties, except those in the message descriptor (or extension), are placed in one or more MQRFH2 headers in the message data.</p> <p>Compatibility. This is the default value; it allows applications which expect JMS related properties to be in an MQRFH2 header in the message data to continue to work unmodified.</p> <p>If the message contains a property with a prefix of mcd . , jms . , usr . , or mqext . then all optional message properties (where the Support value is MQPD_SUPPORT_OPTIONAL), except those in the message descriptor (or extension) are placed in one or more MQRFH2 headers in the message data before the message is sent to the remote queue manager. Otherwise all properties of the message, except those in the message descriptor (or extension), are removed from the message before the message is sent to the remote queue manager.</p> <p>If the message contains a property where the Support field of the property descriptor is not set to MQPD_SUPPORT_OPTIONAL then the message are rejected and treated in accordance with its report options. If the message contains one or more properties where the Support field of the property descriptor is set to MQPD_SUPPORT_OPTIONAL but other fields of the property descriptor are set to non-default values, then these properties are removed from the message before the message is sent to the remote queue manager.</p> <p>None means that all properties of the message, except those in the</p>	<p>PROPCTL</p> <p>Introduction to IBM MQ Explorer 523</p>

Attribute	Meaning	MQSC parameter
Batch data limit	Provide the limit in kilobytes, from 0 - 999999, of the amount of data that should be sent through a channel before taking a sync point. A value of 0 means that no data limit is applied to batches over this channel.	BATCHLIM
Use dead-letter queue	<p>Specifies whether the dead-letter queue is used when messages cannot be delivered by channels. There are two possible values:</p> <ul style="list-style-type: none"> • No means that messages that cannot be delivered by a channel are treated as a failure, and the channel either ends in accordance with the setting of <u>Non-persistent message speed</u>, or discards the messages. • Yes means that if the queue manager <u>Dead-letter queue</u> attribute provides the name of a Dead Letter Queue, then it is used. Otherwise the behavior is as for No. 	USEDLQ

MCA page

The following table lists the attributes on the **MCA** page of the Cluster Queue Manager properties dialog. The attributes show how the Message Channel Agent (MCA) for the selected channel runs.

Attribute	Meaning	MQSC parameter
MCA user ID	The user identifier to be used by the message channel agent when attempting to start a secure LU 6.2 session with a remote message channel agent.	USERID
MCA type	This shows how the message channel agent (MCA) program runs. Thread means that the MCA runs as a thread; Process means that MCA runs as a process.	MCATYPE
MCA name	The Message channel agent name.	MCANAME

Exits page

The following table lists the attributes on the **Exits** page of the Cluster Queue Manager properties dialog. The attributes configure the user exits that are run by the selected channel.

Attribute	Meaning	MQSC parameter
Security exit name	The name of the security exit program.	SCYEXIT

Attribute	Meaning	MQSC parameter
Message exit name	The names of your message exit programs.	MSGEXIT
Send exit name	The names of your send exit programs.	SENDEXIT
Receive exit name	The names of your receive exit programs.	RCVEXIT
Security exit user data	The data that is passed to the channel security exit when the channel security exit is called.	SCYDATA
Message exit user data	The data that is passed to the channel message exit when the channel message exit program is called.	MSGDATA
Send exit user data	The data that is passed to the channel send exit when the send exit program is called.	SENDDATA
Receive exit user data	The data that is passed to the channel receive exit when the receive exit program is called.	RCVDATA

LU6.2 page

The following table lists the attributes on the **LU6.2** page of the Cluster Queue Manager properties dialog.

Attribute	Meaning	MQSC parameter
Mode name	The LU 6.2 mode name.	MODENAME
TP name	The name, or the generic name, of the MCA program that is run at the far end of the link.	TPNAME
User ID	The user identifier that the MCA uses when attempting to initiate a secure LU 6.2 session with a remote MCA.	USERID
Password	The password that the MCA uses when it attempts to initiate a secure LU 6.2 session with a remote MCA.	PASSWORD

Retry page

The following table lists the attributes on the **Retry** page of the Cluster Queue Manager properties dialog. The attributes configure how the channel behaves if the channel cannot connect to the remote queue manager.

Attribute	Meaning	MQSC parameter
Short retry count	The maximum number of times that the channel can try to connect to a remote queue manager.	SHORTRTY

Attribute	Meaning	MQSC parameter
Short retry interval	The approximate interval, in seconds, that the channel must wait before it tries again to connect to the remote queue manager during the short retry count. A value of 0 means that the channel tries again immediately.	SHORTTMR
Long retry count	The maximum number of times that the channel can try to connect to a remote queue manager. The value of this attribute is used only when the count specified in the Short retry count attribute has been exhausted and the channel has still not successfully connected to the remote queue manager.	LONGRTY
Long retry interval	The approximate interval, in seconds, that the channel must wait before it tries again to connect to the remote queue manager during the long retry count. A value of 0 means that the channel tries again immediately.	LONGTMR
Keep alive interval	The value of the Keep alive interval attribute specifies the time-out value of the channel. Auto means that the keepalive value is based on the value of the negotiated Heartbeat interval. If a value is specified and the negotiated Heartbeat interval is greater than zero, the Keep alive interval is the negotiated Heartbeat interval plus 60 seconds; if the negotiated Heartbeat interval is zero, the Keep alive interval is zero too. A value of 0 means that KeepAlive on this channel is disabled.	KAINT

Message retry page

The following table lists the attributes on the **Message retry** page of the Cluster Queue Manager properties dialog. The attributes configure how the channel behaves if the channel fails the first time that it tries to put a message on a remote queue

Attribute	Meaning	MQSC parameter
Message retry count	The number of times that the channel retries to connect to the remote queue manager before it decides that it cannot deliver the message to the remote queue. This attribute controls the action of the MCA only if the Message retry exit name attribute is blank. If the Message retry exit name attribute is not blank, the value of the Message retry count attribute is passed to the exit for the exit's use but the number of times that the channel retries to connect is controlled by the exit, not by the Message retry count attribute.	MRRTY
Message retry interval	The minimum length of time, in milliseconds, that the channel must wait before it can try again to put the message on the remote queue.	MRTMR
Message retry exit name	The name of the channel message-retry exit program.	MRDATA
Message retry exit user data	The data that is passed to the channel message retry exit when the channel message retry exit is called.	MREXIT

Cluster page

The following table lists the attributes on the **Cluster** page of the Cluster Queue Manager properties dialog.

Attribute	Meaning	MQSC parameter
Cluster name	The name of the cluster in which the selected channel definition is shared.	
Cluster queue manager	The name of the queue manager that hosts the selected channel definition.	

Attribute	Meaning	MQSC parameter
Definition type	This shows how the channel was defined. <code>Cluster-sender</code> means that the channel was defined as a cluster-sender channel from an explicit definition; <code>Auto cluster-sender</code> means that the channel was defined as a cluster-sender channel by auto-definition alone; <code>Auto explicit cluster-sender</code> means that the channel was defined as a cluster-sender channel by auto-definition and an explicit definition; <code>Cluster-receiver</code> means that the channel was defined as a cluster-receiver channel from an explicit definition.	DEFTYPE
Queue manager type	This shows the function of the queue manager in the cluster. <code>Repository</code> means that the queue manager is a full repository for the cluster; <code>Normal</code> means that the queue manager is a partial repository for the cluster.	QMTYPE
QMID	The internally generated unique name of the cluster queue manager.	QMID
Network priority	The value of this attribute indicates the channels priority for the network connection; 0 is the lowest priority.	NETPRTY
CLWL channel rank	The rank of the channel in the cluster; 0 is the lowest rank. For more information, see Distributed queuing and clusters in the IBM MQ online product documentation .	CLWLRANK
CLWL channel priority	The priority of the channel in the cluster; 0 is the lowest priority. For more information, see Distributed queuing and clusters in the IBM MQ online product documentation .	CLWLPRTY
CLWL channel weight	The weighting that is applied to the channel so that the proportion of messages that is sent through the channel is controlled; 1 is the lowest weighting. For more information, see Distributed queuing and clusters in the IBM MQ online product documentation .	CLWLWGHT

Attribute	Meaning	MQSC parameter
Transmission queue	<p>The cluster-sender channel is transferring messages from this transmission queue.</p> <p>The name is one of the following transmission queues:</p> <p>SYSTEM.CLUSTER.TRANSMIT.QUEUE</p> <p>The default cluster transmission queue. The queue is shared between cluster-sender channels. The queue is used if the queue manager attribute, Default cluster transmission queue is set to SCTQ, and no transmission queue has its parameter Cluster channel name set to resolve to this cluster-sender channel. The queue is also used if the version of the queue manager is less than IBM WebSphere MQ 7.5.</p> <p>SYSTEM.CLUSTER.TRANSMIT.ChannelName</p> <p>The transmission queue is created by the queue manager, in response to setting the queue manager attribute, Default cluster transmission queue to Queue for each channel. By default, the queue is not shared between cluster-sender channels.</p> <p>User-defined transmission queue</p> <p>The transmission queue parameter, Cluster channel name is manually set to resolve to this cluster-sender channel. Multiple cluster-sender channels might transfer messages from this transmission queue.</p>	XMITQ

SSL page

The following table lists the attributes on the **SSL** page of the Cluster Queue Manager properties dialog. The attributes configure the channel to use TLS security.

Attribute	Meaning	MQSC parameter
SSL CipherSpec	The name of the CipherSpec for an TLS connection. Both ends of the IBM MQ TLS channel definition must have the same value in the SSL CipherSpec attribute.	SSLCIPH
Accept only certificates with Distinguished Names matching these values	The value of the Distinguished Name on the certificate from the peer queue manager or the client at the other end of the IBM MQ channel. When the channel starts, the value of this attribute is compared with the Distinguished Name of the certificate.	SSLPEER
Authentication of parties initiating connections	This parameter specifies how the channel authenticates TLS clients. Required means that the channel must receive and authenticate an TLS certificate from an TLS client; Optional means that the channel is not required to receive and authenticate an TLS certificate from an TLS client. If the value is Optional and the peer TLS client sends a certificate, the channel authenticates the certificate as normal.	SSLCAUTH

Statistics page

The following table lists the attributes on the **Statistics** page of the Cluster Queue Manager properties dialog. The **Statistics** page shows the date and time on which the cluster queue manager was last altered.

Attribute	Meaning	MQSC parameter
Alteration date	The date on which the queue's attributes were last altered.	ALTDATE
Alteration time	The time at which the queue's attributes were last altered.	ALTTIME

Related reference

[“Channel properties” on page 426](#)

You can set properties for all types of channels, including client-connection channels. Some properties are specific to certain types of channel.

[“Cluster queue properties” on page 531](#)

When you view the cluster queues that belong to a queue manager in a cluster, you can double-click the cluster queue and view its properties in the Cluster Queue properties dialog. You cannot edit any of the attributes in the Cluster Queue properties dialog.

Cluster queue properties

When you view the cluster queues that belong to a queue manager in a cluster, you can double-click the cluster queue and view its properties in the Cluster Queue properties dialog. You cannot edit any of the attributes in the Cluster Queue properties dialog.

The following tables list the attributes that are displayed in the Cluster Queue properties dialog. These attributes vary from the attributes shown for the same queues in the Queue properties dialog (see [Queue properties](#)).

- [General](#)
- [Cluster](#)
- [Statistics](#)

For each attribute, there is a brief description of what it means. The tables also give the equivalent MQSC parameter for the DISPLAY QCLUSTER command. For more information about MQSC commands, see [Script \(MQSC\) Commands](#) in the IBM MQ online product documentation.

General page

The following table lists the attributes on the **General** page of the Cluster Queue properties dialog.

Attribute	Meaning	MQSC parameter
Queue name	The name of the cluster queue.	
Queue type	Unlike in the Queue properties dialog, the Queue type attribute just shows that the queue is a cluster queue. See the Cluster page of the Cluster queue properties dialog for whether the cluster queue is a local, remote, or alias queue.	QTYPE
Description	A description of the cluster queue.	DESCR
Put messages	This shows whether queue managers can put messages on the cluster queue. Allowed means that queue managers can put messages on the cluster queue; Inhibited means that queue managers cannot put messages on the cluster queue.	PUT
Default priority	The default priority of messages that are put on the cluster queue, where 9 is the highest priority.	DEFPRTY

Attribute	Meaning	MQSC parameter
Default persistence	This shows whether messages that are put on this cluster queue persist when the queue manager is stopped and restarted. Persistent means that the messages persist; Not persistent means that messages are lost when the queue manager is stopped and restarted.	DEFPSIST

Cluster page

The following table lists the attributes on the **Cluster** page of the Cluster Queue properties dialog. The **Cluster** page shows the attributes of the cluster queue that are relevant to the cluster.

Attribute	Meaning	MQSC parameter
Cluster name	The name of the cluster in which the cluster queue is shared.	CLUSTER
Default bind type	The default message binding.	DEFBIND
Cluster queue type	This is the type of queue that the cluster queue represents: Alias, Local, Queue manager (the cluster queue represents a queue manager alias), Remote queue definition.	CLUSQT
Cluster queue manager	The name of the queue manager that hosts the cluster queue.	CLUSQMGR
QMID	The internally generated unique name of the queue manager that hosts the cluster queue.	QMID
CLWL channel rank	The rank of the queue in the cluster for purposes of cluster workload distribution; 0 is the lowest rank. For more information, see Distributed queuing and clusters in the IBM MQ online product documentation.	CLWLRANK
CLWL channel priority	The priority of the queue in the cluster for purposes of cluster workload distribution; 0 is the lowest priority. For more information, see Distributed queuing and clusters in the IBM MQ online product documentation.	CLWLPRTY

Statistics page

The following table lists the attributes on the **Statistics** page of the Cluster Queue properties dialog. The **Statistics** page shows the date and time on which the cluster queue was last altered.

Attribute	Meaning	MQSC parameter
Alteration date	This is the date on which the queue's attributes were last altered.	ALTDATE
Alteration time	This is the time at which the queue's attributes were last altered.	ALTTIME

Related reference

[“IBM MQ queue properties” on page 412](#)

The attributes that you can set for a queue depend on the type of queue. Different types of IBM MQ queues have different properties. Some of the attributes do not apply to all types of queue, some attributes are specific to cluster queues, and some attributes are specific to z/OS queues.

[“Cluster queue manager properties” on page 517](#)

The **Cluster Queue Manager properties** dialog shows the attributes of the cluster-sender and cluster-receiver channels on the selected cluster queue manager. You cannot edit any of the attributes in the Cluster Queue Manager properties dialog.

Cluster topic properties

You can set attributes for cluster topics. However, you can alter some attributes only while you are creating a new cluster topic. You cannot alter these attributes once the cluster topic has been created.

To configure a standard cluster of queue managers for publish/subscribe messaging, you define one or more administered topic objects on a queue manager in the cluster. To make the topic a cluster topic, you configure the Cluster name property. When you do this, any topic used by a publisher or subscriber, from that point on in the topic tree, is shared across all queue managers in the cluster. Messages published to a clustered branch of the topic tree are automatically routed to subscriptions on other queue managers in the cluster.

General

The following tables list all the attributes for IBM MQ cluster topics. Some of the attributes listed in these tables can only be altered while creating a new topic, and cannot be modified once the IBM MQ cluster topic has been created.

The following table lists the attributes on the **General** page of the **Cluster topic Properties** dialog.

Property	Meaning	MQSC parameter
Topic name	This value cannot be changed once the topic has been created. This parameter is required and cannot contain an empty string. The unique identifier of the administrative topic definition to be created. A maximum of 48 characters are allowed. Name must not be the same as any other topic definition defined on the selected queue manager.	TOPNAME
Topic type	This value is read only. This value displays whether the topic is local; Local, or in a cluster; Cluster.	N/A

Property	Meaning	MQSC parameter
Topic String	<p>This value cannot be changed once the topic has been created. This parameter is required and cannot contain an empty string.</p> <p>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.</p> <p>Topic String must not be the same as any other topic string already represented by another topic object definition. The maximum length of a topic string is 10 240 characters.</p>	TOPICSTR
Description	<p>This value is a string entered by the administrator. It contains descriptive information about the topic. It must contain only displayable characters. A maximum of 64 characters.</p> <p>If characters are used that are not in the coded character set identifier (CCSID) for the selected queue manager, then they might be translated incorrectly if the information is sent to another queue manager.</p>	DESC
Publish	<p>This property controls whether messages can be published to the topic. The default value is As parent. The 2 other options available are:</p> <p>Allowed which means that messages can be published to the topic by an authorized application.</p> <p>Inhibited which means that messages cannot be published to the topic.</p>	PUB
Subscribe	<p>This property controls whether messages can subscribe to the topic. The default value is As parent. The 2 other options available are:</p> <p>Allowed which means that subscriptions can be made to the topic by an authorized application.</p> <p>Inhibited which means that applications cannot subscribe to the topic.</p>	SUB
Durable subscriptions	<p>This property controls whether the topic permits durable subscriptions to be made. The default value is As parent. The 2 other options available are:</p> <p>Allowed which means that durable subscriptions can be made to the topic by an application.</p> <p>Inhibited which means that durable subscriptions cannot be made to the topic by an application.</p>	DURSUB
Default priority	<p>The default priority of messages published to the topic. The default value is As parent.</p> <p>The default priority can be set from 0 (the lowest priority) to 9 (the highest priority)</p>	DEFPTY
Default persistence	<p>The default persistence of a new topic is As parent. Select Persistent to specify that messages created by applications that use MQPER_PERSISTENCE_AS_Q_DEF become persistent. Select Not Persistent to specify that messages created by applications that use MQPER_PERSISTENCE_AS_Q_DEF become non persistent.</p>	DEFPSIS T

Property	Meaning	MQSC parameter
Default put response type	<p>The default response type for message puts. The default value is As parent. The 2 other options available are:</p> <p>Synchronous which means the response is put synchronously.</p> <p>Asynchronous which means the response is put asynchronously.</p>	DEFRESP
Non-persistent message delivery	<p>The delivery method for non-persistent messages published to this topic. The four options are:</p> <p>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. This is the default supplied with IBM MQ, but your installation might have changed it.</p> <p>To all available subscribers 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.</p> <p>To all durable subscribers 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 calls fails.</p> <p>To all subscribers 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 call fails.</p>	NPMMSGDLV
Persistent message delivery	<p>The delivery method for persistent messages published to this topic. The four options are:</p> <p>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. This is the default supplied with IBM MQ, but your installation might have changed it.</p> <p>To all available subscribers 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.</p> <p>To all durable subscribers 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 calls fails.</p> <p>To all subscribers 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 call fails.</p>	PMSGDLV

Property	Meaning	MQSC parameter
Wildcard operation	<p>This value controls the behavior of wildcard subscriptions with respect to the topic. The two values are:</p> <p>Block. Subscriptions made to a wildcard topic less specific than the topic string for this topic object will not receive publications made to this topic or to topic strings more specific than this topic.</p> <p>Passthrough. Subscriptions made to a wildcard topic less specific than the topic string for this topic object will receive publications made to this topic and to topic strings more specific than this topic. This is the default value.</p>	WILDCARD

Distributed Pub/Sub

The following table lists the attributes on the **Distributed Pub/Sub** page of the **Cluster topic Properties** dialog.

Property	Meaning	MQSC parameter
Proxy subscription behavior	<p>Proxy-subscriptions are associated with the queue manager name that created them. Publications are only forwarded to directly connected queue managers if a proxy-subscription exists that includes the publication topic. The two options for this value are:</p> <p>Force. This forces the sending of a wild-carded proxy-subscription for the topic string associated with this topic object from every queue manager in the cluster to every other queue manager in the pub/sub topology, regardless of whether any local subscriptions have been made. Once this forced proxy subscription has been propagated throughout the topology, any new subscriptions immediately receive any publications from other connected queue manager without suffering latency, although all publications are propagated to all other queue managers in the cluster regardless of whether a subscription has requested them or not.</p> <p>Setting this value at a given level in the topic tree also prevents proxy subscriptions being generated for individual topic strings at subsequent levels in the topic tree, reducing the proxy subscription overhead.</p> <p>First use. As the various topologies of the pub/sub queue managers have an interconnected nature, there could be a short delay in the propagation of the proxy-subscription depending on the topology complexity. This means that once a subscription is made, remote publications will not necessarily be received immediately.</p>	PROXYSUB

Property	Meaning	MQSC parameter
Publication scope	<p>The scope of publications can be controlled administratively using the PUBSCOPE topic attribute. The attribute can be set to one of the following 3 values:</p> <ul style="list-style-type: none"> • <code>As parent</code>. This is the default value. The publication scope is set to the same value as the parent queue manager. • <code>Queue manager</code>. The publication is only delivered to local subscribers. • <code>All</code>. The publication is delivered to local subscribers and remote subscribers by directly connected queue managers. 	PUBSCOPE
Subscription scope	<p>The scope of subscriptions can be controlled administratively using the SUBSCOPE topic attribute. The attribute can be set to one of the following 3 values:</p> <ul style="list-style-type: none"> • <code>As parent</code>. This is the default value. The subscription scope is set to the same value as the parent queue manager. • <code>Queue manager</code>. The subscription receives only local publications, and proxy subscriptions are not propagated to remote queue managers. • <code>All</code>. A proxy subscription is propagated to remote queue managers, and the subscriber receives local and remote publications. 	SUBSCOPE

Cluster

The following table lists the attributes on the **Cluster** page of the **Cluster topic Properties** dialog.

Property	Meaning	MQSC parameter
Cluster name	To make a topic a cluster topic, you configure this property. When you do this, any topic used by a publisher or subscriber at that point or in the topic tree is shared across all queue managers in the cluster, and messages published to a clustered branch of the topic tree are automatically routed to subscriptions on other queue managers in the cluster.	CLUSTER
Cluster object state	<p>The current state of this topic object in this cluster. The values can be as follows:</p> <p>ACTIVE The cluster topic is correctly configured and being adhered to by this queue manager.</p> <p>PENDING Only seen by a hosting queue manager, this state is reported when the topic has been created but the full repository has not yet propagated it to the cluster. This might be because the host queue manager is not connected to a full repository, or because the full repository has deemed the topic to be invalid.</p> <p>INVALID This clustered topic definition conflicts with an earlier definition in the cluster and is therefore not currently active.</p> <p>ERROR An error has occurred with respect to this topic object.</p> <p>This parameter is typically used to aid diagnosis when multiple definitions of the same clustered topic are defined on different queue managers, and the definitions are not identical.</p>	CLSTATE
Cluster queue manager	This is the name of the queue manager in the cluster that owns the cluster topic.	N/A

Property	Meaning	MQSC parameter
Cluster route	<p>The routing behavior to use for topics in the cluster defined by the CLUSTER parameter. There are two possible values:</p> <p>DIRECT When you configure a direct routed clustered topic on a queue manager, all queue managers in the cluster become aware of all other queue managers in the cluster. When performing publish and subscribe operations, each queue manager then connects directly to all the others.</p> <p>TOPICHOST When you use topic host routing, all queue managers in the cluster become aware of the cluster queue managers that host the routed topic definitions. When performing publish and subscribe operations, queue managers in the cluster connect only to these topic host queue managers, and not directly to each other. The topic host queue managers are responsible for routing publications from queue managers on which publications are published to queue managers with matching subscriptions.</p>	CLROUTE
QMID	The internally generated unique name of the cluster queue manager. To avoid any ambiguity, it is preferable to use QMID (Queue manager identifier) rather than QMNAME.	QMID

Statistics

The following table lists the attributes on the **Statistics** page of the **Cluster topic Properties** dialog.

Property	Meaning	MQSC parameter
Alteration date	<p>This value cannot be changed, it is provided for information purposes only.</p> <p>This is the date on which the topic's attributes were last altered.</p>	MQCA_ALTERATION_DATE

Property	Meaning	MQSC parameter
Alteration time	<p>This value cannot be changed, it is provided for information purposes only.</p> <p>This is the time at which the topic's attributes were last altered.</p>	MQCA_ALTERATION_TIME

Related tasks

[“Creating and configuring queue managers and objects” on page 27](#)

You can create, configure, and delete queue managers and objects in IBM MQ Explorer by using the Navigator view and Content view.

[“Comparing the properties of two objects” on page 53](#)

You can compare the properties of an object with another object of the same type; for example, compare a queue with another queue, a topic with another topic, or a channel with another channel.

Application connection properties

The attributes for application connections are displayed in the Application Connection properties dialog. You cannot edit the values of any of these attributes.

The following tables list all the attributes for application connections:

- [General](#)
- [Unit of work](#)
- [Handle](#)

For each attribute, there is a brief description of what it means. The tables also give the equivalent MQSC parameter for the DISPLAY CONN command. For more information about MQSC commands, see [Script \(MQSC\) Commands](#) in the IBM MQ online product documentation.

General page

The following table lists the attributes on the **General** page of the Application Connection properties dialog.

Attribute	Meaning	MQSC parameter
Application name	<p>A string containing the tag of the application connected to the queue manager. It is one of the following: a z/OS batch job name, a TSO USERID, a CICS APPLID, an IMS region name, a channel initiator job name, an OS/400 job name, a UNIX process, a Windows process. The application name represents the name of the process or job that has connected to the queue manager. In the instance that this process or job is connected via a channel, the application name represents the remote process or job rather than the local channel process or job name.</p>	APPLTAG

Attribute	Meaning	MQSC parameter
Application type	A string indicating the type of the application that is connected to the queue manager. Batch means an application using a batch connection; RRSBATCH means an RRS-coordinated application using a batch connection; CICS means a CICS transaction; IMS means an IMS transaction; CHINIT means a channel initiator; System means a queue manager; User means a user application.	APPLTYPE
Process ID	The identifier of the process that opened the queue. This attribute is not valid on z/OS.	PID
Thread ID	The identifier of the thread within the application process that has opened the queue. An asterisk indicates that this queue was opened with a shared connection. This attribute is not valid on z/OS.	TID
User ID	The user identifier that is associated with the handle.	USERID
Options	These are the connection options that are currently used by this application connection.	CONNOPTS
Channel name	The name of the channel that owns the handle. If there is no channel associated with the handle, this value is empty. This value is shown only when the handle belongs to the channel initiator.	CHANNEL
Connection name	The connection name that is associated with the channel that owns the handle. If there is no channel associated with the handle, this value is empty. This value is shown only when the handle belongs to the channel initiator.	CONNNAME
PSB name	This is the 8-character name of the program specification block (PSB) that is associated with the running IMS transaction.	PSBNAME
Connection ID	This is the 24-byte unique connection identifier that allows IBM MQ to reliably identify an application. When the application first connects to the queue manager, the queue manager sets the connection identifier.	CONN and EXTCONN

Unit of work page

The following table lists the attributes on the **Unit of work** page of the Application Connection properties dialog. The **Unit of work** page displays information that is available about the unit of work associated with the selected connection.

Attribute	Meaning	MQSC parameter
Unit of work type	The type of unit of recovery as seen by the queue manager. It is one of the following: CICS (z/OS only); XA; RRS (z/OS only); IMS (z/OS only); Queue manager.	URTYPE
Unit of work start date	This is the date that the transaction associated with the connection was started.	UOWSTDA
Unit of work start time	This is the time that the transaction associated with the connection was started.	UOWSTTI
Origin unit of work ID	The unit of recovery identifier, which was assigned by the originator. It is an 8-byte value.	NID
Origin name	This identifies the originator of the thread, except in the case where the Application type attribute is set to RRSBATCH , when it is omitted.	NID
Log extent name	This is the file name of the log extent to which the transaction associated with this connection first wrote.	UOWLOG
First log access date	This is the date that the transaction associated with the connection first wrote to the log.	UOWLOGDA
First log access time	This is the time that the transaction associated with the connection first wrote to the log.	UOWLOGTI

Attribute	Meaning	MQSC parameter
Unit of work state	<p>The state of the unit of work. None means that there is no unit of work; Active means that the unit of work is active; Prepared means that the unit of work is in the process of being committed; Unresolved means that the unit of work is in the second phase of a two-phase commit operation, IBM MQ holds resources on its behalf and external intervention is required to resolve it. This might be as simple as starting the recovery coordinator (such as CICS, IMS, or RRS) or it might involved a more complex operation such as using the RESOLVE INDOUBT command. The Unresolved value can occur only on z/OS.</p>	UOWSTATE
Queue manager unit of work ID	<p>The unit of recovery assigned by the queue manager. On z/OS, this is an 8 byte log RBA, while on other platforms, this is an 8 byte transaction identifier, displayed as 16 hexadecimal characters.</p>	QMURID
External unit of work ID	<p>The external unit of recovery identifier associated with the connection. It is the recovery identifier known in the external syncpoint coordinator. Its format is determined by the value of the UOW type attribute.</p>	EXTURID

Attribute	Meaning	MQSC parameter
Asynchronous state	<p>The state of the asynchronous consumer on this object handle. There are five possible values:</p> <p>Active: An MQCB call has set up a function to call back to process messages asynchronously and the connection handle has been started so that asynchronous message consumption can proceed.</p> <p>Inactive: An MQCB call has set up a function to call back to process messages asynchronously but the connection handle has not yet been started, or has been stopped or suspended, so that asynchronous message consumption cannot currently proceed.</p> <p>Suspended: The asynchronous consumption callback has been suspended so that asynchronous message consumption cannot currently proceed on this object handle. This can be either because an MQCB call with Operation MQOP_SUSPEND has been issued against this object handle by the application, or because it has been suspended by the system. If it has been suspended by the system, as part of the process of suspending asynchronous message consumption the callback function will be called with the reason code that describes the problem resulting in suspension. This will be reported in the Reason field in the MQCBC structure that is passed to the callback function. For asynchronous message consumption to proceed, the application must issue an MQCB call with the Operation parameter set to MQOP_RESUME.</p> <p>Susptemp: The asynchronous consumption callback has been temporarily suspended by the system so that asynchronous message consumption cannot currently proceed on this object handle. As part of the process of suspending asynchronous message consumption, the callback function will be called with the reason code that describes the problem resulting in suspension. This will be reported in the Reason field</p>	<p>ASTATE</p> <p>Introduction to IBM MQ Explorer 545</p>

Attribute	Meaning	MQSC parameter
Units of recovery disposition	<p>(z/OS only) This parameter is used to filter the list of connections returned. There are 3 options to choose from:</p> <ul style="list-style-type: none"> • All means that all connections are returned. This is the default value. • Group means that the connections returned will consist only of those in the group to which the command was targeted. • Queue manager means that the connections returned will consist only of those on the queue manager to which the command was targeted. 	URDISP

Handle page

The following table lists the attributes on the **Handle** page of the Application Connection properties dialog. The **Handle** page displays information about the object that the selected application has opened.

Attribute	Meaning	MQSC parameter
Connection ID	This is the 24-byte unique connection identifier that allows IBM MQ to reliably identify an application. When the application first connects to the queue manager, the queue manager sets the connection identifier.	CONN and EXTCONN
Object name	This is the name of the object that the connection has opened.	OBJNAME
Object type	This is the type of the object that the connection has opened; for example, Queue, Channel, Storage Class.	OBJTYPE

Attribute	Meaning	MQSC parameter
Open options	<p>These are the options used by the connection to open the object.</p> <p>Bind as queue definition means that the application opened the queue to get messages using the queue-defined default; Input shared means that the application opened the queue to get messages with shared access; Input exclusive means that the application opened the queue to get messages with exclusive access; Browse means that the application opened the queue to browse the messages on the queue; Output means that the application opened the queue to put messages on the queue; Inquire means that the application opened the object to get a list of the object's attributes; Set means that the application opened the queue to set the queue's attributes.</p> <p>Bind on open means that application the local queue manager bound the queue handle to a particular instance of the destination queue when the queue was opened so that all messages put using this handle are sent to the same instance of the destination queue, and by the same route; Bind not fixed means that the local queue manager did not bind the queue handle to a particular instance of the destination queue, so successive MQPUT calls using this handle might result in the messages being sent to different instances of the destination queue, or being sent to the same instance but by different routes; Bind as queue default means that the local queue manager bound the queue handle in the way defined by the queue's Default bind type attribute.</p> <p>Save all context means that context information from any message retrieved using this handle is associated with this handle; Pass identity context means that the identity context information from a message can be passed to the processed message when it is put on the queue; Pass all context means that the identity and origin context information</p>	<p>OPENOPTS</p>

Attribute	Meaning	MQSC parameter
QSG disposition	Read-only. The queue sharing group disposition of the object. Queue manager means that the object definition is available only to the queue manager that hosts it; Group means that the object definition is stored on the shared repository and each queue manager in the queue sharing group has a copy of the definition; Copy means that the object definition is the queue manager's copy of a definition in the shared repository; Shared means that the object definition is stored on the queue sharing group's coupling facility and is available to all the queue managers in the queue sharing group.	QSGDISP
Handle state	The current state of the handle. Active means that an API call from this connection is currently in progress for this object. If the object is a queue, this condition can arise when an MQGET WAIT call is in progress. If there is an MQGET signal outstanding, this does not mean, by itself, that the handle is active. Inactive means that no API call from this connection is currently in progress for this object. If the object is a queue, this condition can arise when no MQGET WAIT call is in progress.	HSTATE
Topic string	The resolved topic string. This parameter is relevant for handles with OBJTYPE(TOPIC). For any other object type, this parameter is blank.	TOPICSTR
Subscription name	The application's unique subscription name that is associated with the handle. This parameter is relevant only for handles of subscriptions to topics. Not all subscriptions will have a subscription name.	SUBNAME

Attribute	Meaning	MQSC parameter
Subscription ID	The internal, all-time unique identifier of the subscription. This parameter is relevant only for handles of subscriptions to topics. Not all subscriptions show up in DISPLAY CONN; only those that have current handles open to the subscription show up. You can use the DISPLAY SUB command to see all subscriptions.	SUBID
Destination queue manager	The destination queue manager for messages that are published to this subscription. This parameter is relevant only for handles of subscriptions to topics. If DEST is a queue that is hosted on the local queue manager, this parameter will contain the local queue manager name. If DEST is a queue that is hosted on a remote queue manager, this parameter will contain the name of the remote queue manager.	DESTQMGR
Destination name	The destination queue for messages that are published to this subscription. This parameter is only relevant for handles of subscriptions to topics.	DEST

Attribute	Meaning	MQSC parameter
Asynchronous state	<p>The state of the asynchronous consumer on this object handle. There are five possible values:</p> <p>Active: An MQCB call has set up a function to call back to process messages asynchronously and the connection handle has been started so that asynchronous message consumption can proceed.</p> <p>Inactive: An MQCB call has set up a function to call back to process messages asynchronously but the connection handle has not yet been started, or has been stopped or suspended, so that asynchronous message consumption cannot currently proceed.</p> <p>Suspended: The asynchronous consumption callback has been suspended so that asynchronous message consumption cannot currently proceed on this object handle. This can be either because an MQCB call with Operation MQOP_SUSPEND has been issued against this object handle by the application, or because it has been suspended by the system. If it has been suspended by the system, as part of the process of suspending asynchronous message consumption the callback function will be called with the reason code that describes the problem resulting in suspension. This will be reported in the Reason field in the MQCBC structure that is passed to the callback function. For asynchronous message consumption to proceed, the application must issue an MQCB call with the Operation parameter set to MQOP_RESUME.</p> <p>Susptemp: The asynchronous consumption call back has been temporarily suspended by the system so that asynchronous message consumption cannot currently proceed on this object handle. As part of the process of suspending asynchronous message consumption, the callback function will be called with the reason code that describes the problem resulting in suspension. This will be reported in the Reason field in the MQCBC structure passed to the callback function. The</p>	ASTATE

Attribute	Meaning	MQSC parameter
Read ahead	<p>The read ahead connection status. There are four possible values:</p> <p>No: Read ahead of non-persistent messages is not enabled for this object. This is the default value.</p> <p>Yes: Read ahead of non-persistent message is enabled for this object and is being used efficiently.</p> <p>Backlog: Read ahead of non-persistent messages is enabled for this object. Read ahead is not being used efficiently because the client has been sent a large number of messages which are not being consumed.</p> <p>Inhibited: Read ahead was requested by the application but has been inhibited because of incompatible options specified on the first MQGET call.</p>	READA

Related tasks

“Viewing and closing connections to applications” on page 203

You can use the **Application Connections** dialog to find out which applications are currently connected to a specific queue manager, and which queue manager objects an application is currently accessing. You can also use this dialog to close a connection.

Message properties

Message attributes are displayed in the Message properties dialog. You cannot edit any of the message attributes.

The following tables list the attributes of IBM MQ messages that you can put and get from queues:

- [General](#)
- [Report](#)
- [Context](#)
- [Identifiers](#)
- [Segmentation](#)
- [Named Properties](#)
- [MQRFH2 Properties](#)
- [Data](#)
- [Dead-letter header](#)

For each attribute, there is a brief description of meaning of the attribute. The tables also show the MQMD form of the name, as used in the API. This is described in [Overview for MQMD](#) in the IBM MQ online product documentation.

General page

The following table lists the attributes on the **General** page of the Message properties dialog.

Attribute	Meaning	MQMD form
Position	Read-only. The current position in the queue of the message.	(Not applicable.)
Message type	Read-only. This is the type of the message: Datagram means that the message does not require a reply; Request means that the message requires a reply; Reply means that the message is a reply to an earlier request message; Report means that the message is reporting on some expected or unexpected occurrence, usually related to some other message. For example, a request message that contained data that was not valid.	MsgType
Priority	Read-only. This is the priority of the message. The lowest priority is 0.	Priority
Persistence	Read-only. This indicates whether the message is persistent or nonpersistent. If the message is persistent, it survives system failures and restarts of the queue manager. If the message is nonpersistent, it survives a restart only if it is present on a queue having the NPMCLASS(HIGH) attribute. However, even with the NPMCLASS(HIGH) attribute a message does not survive a QMGR class. Nonpersistent messages on queues having the NPMCLASS(NORMAL) attribute are discarded at queue manager restart, even if the message is found on the auxiliary storage during the restart procedure.	Persistence
Put date/time	Read-only. This is the date when the message was put.	PutDate; PutTime
Expiry	Read-only. This is the period of time, in tenths of a second, after which the message becomes eligible to be discarded if it has not already been removed from the target queue. The expiry interval is set by the application that put the message.	Expiry
Reply-to queue	Read-only. This is the name of the message queue to which the application that issued the get request for the message should send the reply and report messages.	ReplyToQ

Attribute	Meaning	MQMD form
Reply-to queue manager	Read-only. This is the name of the queue manager on which the reply-to queue is defined.	ReplyToQmgr
Backout count	Read-only. This is the number of times the message has previously been returned by the MQGET call as part of a unit of work, and subsequently backed out.	BackoutCount

Report page

The following table lists the attributes on the **Report** page of the Message properties dialog. A report is a message about another message, used to inform the application about expected or unexpected events that relate to the original message. The **Report** page displays the attributes related to report messages. For more information, see [Report options and message flags](#) in the IBM MQ online product documentation.

Attribute	Meaning	MQMD form
Report	Read-only. This field is where the sender application specifies whether report messages are required, whether the application data is to be included in the report messages, and also how the message and correlation identifiers in the report or reply message are to be set.	Report
Feedback	Read-only. This is used only with report messages to indicate the nature of the report.	Feedback
Original length	Read-only. This is used only with report messages to indicate the length of the original message to which the report relates.	OriginalLength

Context page

The following table lists the attributes on the **Context** page of the Message properties dialog. The **Context** page displays information from the sender application about the message.

Attribute	Meaning	MQMD form
User identifier	Read-only. This is the user identifier of the application that originated the message.	UserIdentifier
Application type	Read-only. This is the type of application that put the message, for example, CICS or AIX.	PutApplType
Put application name	Read-only. This is the name of the application that put the message. This name might appear truncated.	PutApplName

Attribute	Meaning	MQMD form
Application identity data	Read-only. This is information that is defined by the application suite, and can be used to provide information about the message or its originator.	ApplIdentityData
Application origin data	Read-only. This is information that is defined by the application suite, and can be used to provide additional information about the origin of the message.	ApplOriginData
Accounting token	Read-only. This is information that allows the application to appropriately charge work that is done as a result of the message.	AccountingToken

Identifiers page

The following table lists the attributes on the **Identifiers** page of the Message properties dialog. The **Identifiers** page displays identification information that is associated with the message.

Attribute	Meaning	MQMD form
Message identifier	Read-only. This is the message identifier, which is used to distinguish one message from another.	MsgId
Message identifier bytes	Read-only. This is the message identifier in byte form.	MsgId
Correlation identifier	Read-only. This is the correlation identifier, which the application can use to relate one message to another, or to relate the message to other work that the application is performing.	CorrelId
Correlation identifier bytes	Read-only. This is the correlation identifier in byte form.	CorrelId
Group identifier	Read-only. This is the group identifier, which is used to identify the particular message group or logical message to which the physical message belongs.	GroupId
Group identifier bytes	Read-only. This is the group identifier in byte form.	GroupId

Segmentation page

The following table lists the attributes on the **Segmentation** page of the Message properties dialog. The **Segmentation** page displays the attributes related to segmenting large messages.

Attribute	Meaning	MQMD form
Logical sequence number	Read-only. This is the sequence number of the logical message within the group. Sequence numbers start at 1, and increase by 1 for each new logical message in the group, up to a maximum of 999,999,999. A physical message that is not in a group has a sequence number of 1.	MsgSeqNumber
Offset	Read-only. This is the offset of data in the physical message from the start of the logical message.	Offset
Flags	Read-only. These are the message flags that specify attributes of the message, or control its processing.	MsgFlags

Named Properties page

The following table lists the attributes on the **Named Properties** page of the **Message properties** dialog. The **Named Properties** page is present only when you have selected as **Named Properties** on the **Message properties** page of the **Preferences** dialog, and if the selected message has properties. Properties of the message, except those contained in the message descriptor or extension, must be represented in the **Named Properties** panel in name value pairs, and the properties are removed from the message data.

The MQGMO Option is **MQGMO_PROPERTIES_IN_HANDLE**, for more information on the relevant Get Message Options see [“MQ Get Message Options” on page 557](#).

For more information on the **Preferences** dialog, see [Configuring MQ Explorer](#).

Attribute	Meaning
Name	Read-only. The name of the message property.
Value	Read-only. This is the actual value of the named property.

MQRFH2 Properties page

The following table lists the attributes on the **MQRFH2 Properties** page of the **Message properties** dialog. There are 2 ways to make the **MQRFH2 Properties** page visible:

- Select as an MQRFH2 structure in message body on the **Messages** page of the **Preferences** dialog.

Properties of the message, except those contained in the message descriptor or extension, must be represented in the **MQRFH2 Properties** and the properties remain in the message data. The MQGMO Option is **MQGMO_PROPERTIES_FORCE_MQRFH2**, for more information on the relevant Get Message Options see [“MQ Get Message Options” on page 557](#).

- Select as an MQRFH2 structure in message body compatible with WebSphere MQ V6 on the **Messages** page of the **Preferences** dialog. The **MQRFH2 Properties** page will only be visible if the message contains a property with a prefix of *mcd*, *jms*, *usr*, or *mqext*

If the message contains a property with a prefix of *mcd*, *jms*, *usr*, or *mqext*, all message properties, except those contained in the message descriptor or extension, must be represented in the MQRFH2 Properties panel and the properties remain in the message data. Otherwise, all properties of the message, except those contained in the message descriptor or extension, are discarded and not

displayed. The MQGMO Option is **MQGMO_PROPERTIES_IN_COMPATIBILITY**, for more information on the relevant Get Message Options see “MQ Get Message Options” on page 557.

For more information on the **Preferences** dialog, see [Configuring MQ Explorer](#).

As the MQRFH2 structure is nested xml, the **MQRFH2 Properties** page displays the MQRFH2 properties in a tree view. All properties with the same synonym are grouped under the synonym tree which can be expanded to display all the properties, and collapsed to hide all the properties.

Attribute	Meaning
Name	Read-only. The name of the message property.
Value	Read-only. This is the actual value of the named property.

Data page

The following table lists the attributes on the **Data** page of the Message properties dialog. The **Data** page displays the message data itself and information about the data format.

Attribute	Meaning	MQMD form
Data length	Read-only. This is the length of the original message.	OriginalLength
Format	Read-only. This is the name that the sender of the message has used to indicate to the receiver the nature of the data in the message.	Format
Coded character set identifier	Read-only. This is the coded character set identifier of the character data in the application message data.	CodedCharSetId
Encoding	Read-only. This is the numeric encoding of numeric data in the message. This value does not apply to numeric data in the MQMD structure itself.	Encoding
Message data	Read-only. This is the message data in human readable ASCII text.	(Not applicable.)
Message data bytes	Read-only. This is the message data in hexadecimal format.	(Not applicable.)

Dead-letter header page

The following table lists the attributes on the **Dead-letter header** page of the Message properties dialog. The **Dead-letter header** page is present only when the message has a dead-letter header.

Attribute	Meaning	MQMD form
Reason	This identifies the reason why the message was placed on the dead-letter (undelivered message) queue instead of on the original destination queue.	Reason

Attribute	Meaning	MQMD form
Destination queue	The name of the message queue that was the original destination for the message.	DestQName
Destination queue manager	The name of the queue manager that was the original destination for the message.	DestQMgrName
Original encoding	This specifies the numeric encoding of the data that follows the MQDLH structure (usually the data from the original message); it does not apply to numeric data in the MQDLH structure itself.	Encoding
Original CCSID	This specifies the character set identifier of the data that follows the MQDLH structure (usually the data from the original message); it does not apply to character data in the MQDLH structure itself.	CodedCharSetId
Original format	This specifies the format name of the data that follows the MQDLH structure (usually the data from the original message).	Format
Put application type	The type of application that put the message. This is part of the origin context of the message. For more information on message context, see Message context in the IBM MQ online product documentation.	PutApplType
Put application name	The name of the application that put the message on the dead-letter queue. The format of the name depends on the Put Application Type attribute. This name might appear truncated.	PutApplName
Put date	The date when the message was put on the dead-letter queue.	PutDate
Put time	The time when the message was put on the dead-letter queue.	PutTime

Related tasks

[“Sending test messages” on page 89](#)

You can use a test message to check whether an application or a queue manager can put a message on a queue. You can also browse messages that are already on a queue or clear messages from a queue.

MQ Get Message Options

MQ Get Message options control the action of MQGET.

You can specify none, one, or more of the options described later in this section. If you need more than one option, the values can be:

- Added (do not add the same constant more than once), or

- Combined using the bitwise OR operation (if the programming language supports bit operations).

The initial value of the `Options` field is `MQGMO_NO_WAIT` plus `MQGMO_PROPERTIES_AS_Q_DEF`.

Property options

The following options relate to the properties of the message:

MQGMO_PROPERTIES_AS_Q_DEF

Properties of the message, except the properties contained in the message descriptor (or extension) must be represented as defined by the **PropertyControl** queue attribute. If a *MsgHandle* is provided this option is ignored and the properties of the message are available using the *MsgHandle*, unless the value of the **PropertyControl** queue attribute is `MQPROP_FORCE_MQRFH2`.

This is the default action if no property options are specified.

MQGMO_PROPERTIES_IN_HANDLE

Properties of the message must be made available using the *MsgHandle*. If no message handle is provided the call fails with reason **MQRC_HMSG_ERROR**.

MQGMO_NO_PROPERTIES

No properties of the message, except the properties contained in the message descriptor (or extension) are retrieved. If a *MsgHandle* is provided it is ignored.

MQGMO_PROPERTIES_FORCE_MQRFH2

Properties of the message, except the properties contained in the message descriptor (or extension) must be represented using **MQRFH2** headers. This provides compatibility with earlier versions for applications which are expecting to retrieve properties but are unable to be changed to use message handles. If a *MsgHandle* is provided it is ignored.

MQGMO_PROPERTIES_COMPATIBILITY

If the message contains a property with a prefix of **"mcd."**, **"jms."**, **"usr."**, or **"mqext."**, all message properties are delivered to the application in an **MQRFH2** header. Otherwise all properties of the message, except the properties contained in the message descriptor (or extension), are discarded and are no longer accessible to the application.

Default option

If none of the options described previously is required, the following option can be used:

MQGMO_NONE

Use this value to indicate that no other options have been specified; all options assume their default values. `MQGMO_NONE` aids program documentation; it is not intended that this option is used with any other, but because its value is zero, such use cannot be detected.

Connection details properties

You can view and set properties for connections in the Connection Details properties dialog.

To view connection details properties, right-click on the queue manager, select **Connection details**, and then click **Properties**. The following tables list all the properties that you can set:

- [General](#)
- [Security exit](#)
- [Userid](#)
- [SSL key repositories](#)
- [SSL options](#)

General page

The following table lists the properties that you can set on the **General** page of the Connection Details properties dialog.

Item	Description
Queue manager name	Read-only. The name of the local queue manager.
Connection type	Read-only. The type of connection. The three possible values are: <ol style="list-style-type: none">1. Local. A local connection.2. Client. A client connection.3. Indirect. A connection through another queue manager.
Connection name	Read-only. The connection name that is associated with the channel that owns the handle. If no channel is associated with the handle, this value is empty. This value is shown only when the handle belongs to the channel initiator.
Channel name	Read-only. The name of the channel definition.
Refresh interval (seconds)	The frequency with which IBM MQ Explorer refreshes its information about the queue manager.
Autoreconnect	The configuration of the connection to each queue manager so that IBM MQ Explorer automatically connects to it at startup or reconnects to it if the connection is lost; for example, if the network connection to a remote queue manager fails. Important: You can enable either the Autoreconnect property, or the Prompt for password property, described on the “Userid page” on page 560, but not both.
Wait interval	The number of seconds that IBM MQ Explorer waits for messages on the reply queue.
Expiry interval	The number of seconds that messages remain on the reply queue. This queue is the one that IBM MQ Explorer is using to communicate with the queue manager.

Security exit page

The following table lists the properties that you can set on the **Security exit** page of the Connection Details properties dialog.

Item	Description
Exit name	Specifies the name of the exit program to be run by the security exit. Exit name can be up to 1024 characters long and is case sensitive. Exit name can be a fully qualified java class name found in the directory or jar file. Exit name can be a C exit, of the format: <code>dll_name(function_name)</code> . The default path for exits is always used to locate C exits, you cannot specify the location of the exit library in this entry field unless no default path is set.
in directory	Specifies the directory for the security exit (Java exits only).

Item	Description
in jar	Specifies the jar file for the security exit (Java exits only).
Exit data	Exit data can be up to 32 characters long. If no value has been defined for that attribute, this field is all blanks.

Userid page

The following table lists the properties that you can set on the **Userid** page of the Connection Details properties dialog.

Item	Description
Enable user identification	Select Enable user identification to enable the fields on this dialog.
User identification compatibility Mode	When selected, the userid and password are passed to the server in a way compatible with security exits created prior to IBM MQ 8.0.
Userid	The userid and password, when specified, are passed to the server, and can be used either by: <ul style="list-style-type: none"> • The queue manager, if configured to use connection authentication, or • A server security exit, if using a client connection to establish the identity of the IBM MQ Explorer user.
No password	When selected, no password is passed to the server with the userid.
Prompt for password	When selected, the user is prompted for a password that is passed to the server with the userid. The prompting occurs as part of the connect operation. Important: You can enable either the Prompt for password property, or the Autoreconnect property, described on the “General page” on page 559, but not both.
Use saved password	When selected, the saved password is passed to the server with the userid.
Saved password	The saved password to be passed to the server with the userid.

SSL key repositories page

The following table lists the properties that you can set on the **SSL key repositories** page of the Connection Details properties dialog.

Item	Description
Trusted Certificate Store	The location of the truststore on the computer. In the Trusted Certificate Store field, browse for the location of the truststore on the computer. The truststore and keystore contain the TLS certificates that are used with connections that use client channel definition tables. It is possible that the truststore and keystore are in the same location on your computer.

Item	Description
Personal Certificate Store	The location of the truststore on the computer. In the Personal Certificate Store field, browse for the location of the keystore on the computer.

For more information about configuring IBM MQ Explorer with the default location and password of the TLS certificate store, see [“Specifying the default location and default password of TLS certificates” on page 105](#).

SSL options page

The following table lists the properties that you can set on the **SSL options page** of the Connection Details properties dialog.

Item	Description
SSL FIPS required	Read-only. If set to No (the default), any available cipher suite can be used. If set to Yes , then only FIPS-certified cipher suites can be used.
SSL CipherSpec	<p>The SSL CipherSpec identifies the combination of encryption algorithm and hash function that is used by a TLS connection. A CipherSpec forms part of a cipher suite, which identifies the key exchange and authentication mechanism and the encryption and hash function algorithms.</p> <p>The size of the key that is used during the TLS handshake can depend on the digital certificate you use, but some of the CipherSpecs supported by IBM MQ include a specification of the handshake key size. Larger handshake key sizes provide stronger authentication. With smaller key sizes, the handshake is faster.</p>
SSL reset count	The number of bytes, 0 - 999 999 999, that are sent and received within a TLS conversation before the secret key is renegotiated. A value of 0 means that the secret key is never renegotiated. The number of bytes includes control information that is sent by the message channel agent (MCA). If the value of this attribute is greater than 0 and the value of the Heartbeat interval attribute in the Channel properties is greater than 0, the secret key is also renegotiated before message data is sent or received following a channel heartbeat.
Peer name	The Distinguished Name (DN) of the queue manager to be used by TLS. The peer name is set to indicate that connections are allowed only where the server is successfully authenticated as a specific DN.

Connection factory properties

You can view and set connection factory properties in the Connection Factory properties dialog. The properties that are available in the dialog depend on which messaging provider the connection factory uses.

The following tables list all the properties that you can set:

- [General](#)
- [Connection](#)
- [Reconnection](#)
- [Channels](#)
- [SSL](#)
- [Exits](#)
- [Broker](#)
- [Temporary queues](#)
- [Temporary topics](#)
- [Subscriber](#)
- [Extended](#)
- [Advanced tuning](#)

For each property, there is a brief description of when you might need to configure the property. The tables also give the equivalent long and short names to use in the JMS Administration command line tool. The properties that are available in the Properties dialog depend on which messaging provider the connection factory uses. For more information about the JMS Administration command line tool, see [Configuring JMS objects using the administration tool](#) in the online product documentation.

General page

The following table lists the properties that you can set on the **General** page of the Connection Factory properties dialog.

Property	Meaning	Long name	Short name
Name	This is the name of the object.	NAME	
Description	Type a description of the object.	DESCRIPTION	DESC
Class name	This shows the class name that is implemented by the connection factory.		
Messaging provider	This shows whether the connection factory uses MQ transport (Bindings or Client) or Real-time transport (Direct or Direct HTTP). Real-time transport is not available in IBM MQ 8.0. If you are using IBM MQ 8.0 you can define Real-time transport, but it fails when an attempt is made to create a connection.		

Property	Meaning	Long name	Short name
Transport	<p>This shows the transport type used for the connection. Bindings is a direct connection to the queue manager that is on the same computer as the JMS client; Client is a client connection using TCP/IP (the queue manager can be on the same or different computer); Direct is a direct connection to a broker of IBM Integration Bus; Direct HTTP is a direct connection using HTTP tunneling. Although this field is read-only, you can change the transport type if there is an alternative transport type available for the messaging provider; for more information, see Changing the transport type used for connections.</p>	TRANSPORT	TRAN

Property	Meaning	Long name	Short name
Provider version	<p>Select, or type, the version, release, modification level and fix pack of the queue manager to which this client is intended to connect. If you type the value, use one of the following formats:</p> <ul style="list-style-type: none"> • V.R.M.F • V.R.M • V.R • V <p>where V, R, M and F are integer values greater than or equal to zero.</p> <p>A value of "8" or greater indicates that this is intended as an IBM MQ 8.0 ConnectionFactory for connections to an IBM MQ 8.0 queue manager. A value of "7" or greater indicates that this is intended as an IBM WebSphere MQ 7.0 ConnectionFactory for connections to an IBM WebSphere MQ 7.0 queue manager. A value less than 7 (for example "6.0.2.0"), indicates that it is intended for use with queue managers earlier than IBM WebSphere MQ 7.0. The default value, unspecified, allows connections to any level of queue manager, determining the applicable properties and functionality available based on the queue manager's capabilities.</p> <p>For more information about troubleshooting problems that can arise when you are connecting to a queue manager with a specific provider version, see JMS provider version troubleshooting in the IBM MQ online product documentation.</p>	PROVIDERVERSION	PVER

Property	Meaning	Long name	Short name
Client identifier	The client identifier is used to uniquely identify the application connection for durable subscriptions. Type an identifier for the client	CLIENTID	CID
Max buffer size	Type the maximum number of received messages that can be stored in an internal message buffer while waiting to be processed by the client application. This property applies only when the Transport property has the value Direct and Direct HTTP. The default is 1000.	MAXBUFFSIZE	MBSZ

Connection page

The following table lists the properties that you can set on the **Connection** page of the Connection Factory properties dialog. Edit the properties on the **Connection** page to set the connection details for connections created by this connection factory.



Property	Meaning	Long name	Short name
Base queue manager	Select or type the name of the queue manager to connect to. If your application uses a client channel definition table to connect to a queue manager, see Client channel definition table in the IBM MQ online product documentation.	QMANAGER	QMGR
Broker queue manager	This property can be used in an IBM WebSphere MQ 7.0 JMS client but has no effect on an IBM WebSphere MQ 7.0 queue manager. Select or type the name of the queue manager on which the broker is running.	BROKERQMGR	BQM

Property	Meaning	Long name	Short name
Connection list	Comma separated list of HOSTNAME(PORT) connection addresses. The list is tried in order, once per each connection retry attempt. HOSTNAME can be a DNS name or IP address. PORT defaults to 1414.	CLIENTRECONNECTHOSTS	CRHOSTS
Host name	Type the host name or IP address of the computer that hosts the queue manager or, for a direct connection to a broker, the computer that hosts the broker.	HOSTNAME	HOST
Port	Type the port number on which the queue manager or broker listens. The default is 1414 if the Transport property is set to Client; the default is 1506 if the Transport property is set to Direct or Direct HTTP.	PORT	-
Proxy host name	Type the host name of the proxy server for a direct connection.	PROXYHOSTNAME	PHOST
Proxy port	Type the port number of the proxy server for a direct connection. The default is 443.	PROXYPORT	PPORT
Coded character set ID	Type the coded character set ID (CCSID) to be used on connections. For optimum performance, the value of this property should be the same as the value of the Coded Character Set ID attribute (Queue manager properties) of the base queue manager.	CCSID	CCS

Property	Meaning	Long name	Short name
Multicast	<p>Specify whether messages are delivered to message consumers using multicast transport. Multicast transport is applicable only to topic destinations and can be used only when the connection factory uses Direct IP transport. Direct IP transport is not available in IBM MQ 8.0. If you are using IBM MQ 8.0 you can define Direct IP transport, but it fails when an attempt is made to create a connection.</p> <p>The default value is Disabled, which means that messages are not delivered to a message consumer using multicast transport.</p> <p>Click Enabled to deliver messages to the message consumer using multicast transport. The topic must be configured for reliable multicast in the broker; a reliable quality of service is used if the topic is configured for reliable multicast.</p> <p>Click Reliable to deliver messages to the message consumer using multicast transport with a reliable quality of service. The topic must be configured for reliable multicast in the broker; if the topic is not configured for reliable multicast, you cannot create a message consumer for the topic.</p> <p>Click Not reliable to deliver messages using multicast transport but without using a reliable quality of service. The topic must be configured for multicast in the broker; a reliable quality of service is not used, even if the topic is configured for reliable multicast.</p>	MULTICAST	MULTI

Property	Meaning	Long name	Short name
Local address	<p>For a connection to a queue manager, specify either the local network interface to be used, or the local port (or range of local ports) to be used, or both. The channel binds to the address locally. Use the format <i>ipaddress(low-port, high-port)</i>, where <i>ipaddress</i> is the IP address specified in IPv4 dotted decimal, IPv6 hexadecimal, or alphanumeric host name format. For example, <i>127.0.0.1</i> specifies the IPv4 address with any port; <i>127.0.0.1(1000)</i> specifies the IPv4 address and a specific port; <i>127.0.0.1(1000,2000)</i> specifies the IPv4 address and a range of ports; <i>(1000)</i> specifies a port only. If you are using a direct connection to a broker, this property is relevant only when multicast is used. Specify the local network interface to be used, as an IP address or as a host name, but do not specify any port number.</p>	LOCALADDRESS	LA

Property	Meaning	Long name	Short name
Connect options	<p>Select how the application connects to the queue manager. Click Standard for the nature of the binding between the application and the queue manager to depend on the platform on which the queue manager is running and how the queue manager is configured. Click Shared for the application and the local queue manager agent to run in separate units of execution but share some resources. Click Isolated for the application and the local queue manager agent to run in separate units of execution. Click Fastpath for the application and the local queue manager agent to run in the same unit of execution. Click Serial queue manager for the application to request exclusive use of the connection tag within the scope of the queue manager. Click Serial queue sharing group for the application to request exclusive use of the connection tag within the scope of the queue-sharing group to which the queue manager belongs. Click Restricted queue manager for the application to request shared use of the connection tag but restricting the shared use of the connection tag within the scope of the queue manager. Click Restricted queue-sharing group for the application to request the shared use of the connection tag but restricting the shared use of the connection tag within the scope of the queue sharing group to which the queue manager belongs. The Standard, Shared, Isolated, and</p>	CONNOPT	CNOPT

Property	Meaning	Long name	Short name
  Connection tag	This is a tag that the queue manager associates with the resources that are updated by the application within a unit of work while the application is connected to the queue manager. The connection tag is supported only by IBM MQ for z/OS queue managers.	CONNTAG	CNTAG
Share conversations allowed	Click Yes (the default value) to specify that a client connection can share its socket with other JMS connections from the same process to the same queue manager, if the channel definitions match. Otherwise, click No .	SHARECONVALLOWED	SCA

Reconnection page

Property	Meaning	Long name	Short name
Options	<p>Do not reconnect The application will not be reconnected.</p> <p>Reconnect The application can reconnect to any queue manager. Use the reconnect option only if there is no affinity between the client application and the queue manager with which it initially established a connection.</p> <p>Reconnect to the same queue manager The application can reconnect, but only to the queue manager to which it originally connected. Use this value if a client can be reconnected, but there is an affinity between the client application, and the queue manager to which it first established a connection. Choose this value if you want a client to automatically reconnect to the standby instance of a highly available queue manager. Automatic client reconnect is not supported by IBM MQ classes for Java.</p>	CLIENTRECONNECTOPTI ONS	CROPT
Timeout	Interval in seconds before reconnection retries cease. The default is 1800 seconds (30 minutes).	CLIENTRECONNECTTIME OUT	CRT

Channels page

The following table lists the properties that you can set on the **Channel** page of the Connection Factory properties dialog. Edit the properties on the **Channel** page to configure how the JMS client connects to the queue manager.

Property	Meaning	Long name	Short name
Channel	Select or type the name of the server-connection channel to use to connect the client to the queue manager. To use Select to enter the name, you must have already entered a value for the Base Queue Manager property on the Connection page, and the selected queue manager must be running. You must set a value for either the Channel property or for the Client Channel Definition Table URL property but not both.	CHANNEL	CHAN
Client channel definition table URL	Enter the uniform resource locator (URL) that identifies the name and location of the file that contains the client channel definition table and specifies how the file can be accessed. You must set a value for either the Channel property or for the Client Channel Definition Table URL property but not both.	CCDTURL	CCDT
Header compression	Click Edit then select the list of techniques that can be used for compressing header data on a connection.	COMPHDR	HC
Message compression	Click Edit then select the list of techniques that can be used for compressing message data on a connection.	COMPMSG	MC

SSL page

The following table lists the properties that you can set on the **SSL** page of the Connection Factory properties dialog. Edit the properties on the **SSL** page to configure the TLS details for securing client connections and direct connections to the broker.

Property	Meaning	Long name	Short name
Direct SSL authentication	To enable TLS authentication for a direct connection, click Certificate . To disable authentication, user name authentication, and password authentication, click Basic .	DIRECTAUTH	DAUTH
CipherSuite	Select the CipherSuite to use for an TLS connection. The CipherSuite must match the CipherSpec of the server-connection channel that is specified in the Channel property on the Channels page . If you do not select a CipherSuite, the FIPS Required, Certificate Revocation List, Peer Name, and Reset Count properties are ignored.	SSLCIPHERSUITE	SCPHS
FIPS required	Click Yes to specify that an TLS connection must use a CipherSuite that is supported by the IBM Java JSSE FIPS provider (IBMJSEFIPS). Click No if an TLS connection can use any CipherSuite.	SSLFIPSREQUIRED	SFIPS
Certificate revocation list	Enter a list of certificate revocation list servers to check for TLS certificate revocation.	SSLCRL	SCRL
Peer name	Type a <i>distinguished name</i> skeleton that must match that provided by the queue manager. The TLS peer name must also match the TLS peer name of the server-connection channel that is specified in the Channel property on the Channels page .	SSLPEERNAME	SPEER
Reset count	Enter the total number of bytes sent and received by a connection before the secret key that is used for encryption is negotiated.	SSLRESETCOUNT	SRC

Exits page

The following table lists the properties that you can set on the **Exits** page of the Connection Factory properties dialog. User exits are code modules that you provide yourself to do additional processing on the message data (for example, encryption or data compression). Edit the properties on the **Exits** page to change the default location of client exit code modules that are run when called.

Property	Meaning	Long name	Short name
Send exit name	Enter the name of the channel send exit or the sequence of send exits to be run in succession. Each entry in the list must be either the name of a class that implements the IBM MQ Java interface MQSendExit (for a channel send exit written in Java), or a string in the format <i>libraryName(entryPointName)</i> (for a channel send exit not written in Java).	SENDEXIT	SDX
Send exit initialization	Enter the user data that is passed to channel send exits when they are called. You can enter one or more items of user data separated by commas.	SENDEXITINIT	SDXI
Receive exit name	Enter the name of the channel receive exit, or a sequence of receive exits, to be called. Each entry in the list must be either the name of a class that implements the IBM MQ Java interface MQReceiveExit (for a channel receive exit written in Java), or a string in the format <i>libraryName(entryPointName)</i> (for a channel receive exit not written in Java).	RECEXIT	RCX
Receive exit initialization	Enter the user data that is passed to channel receive exits when they are called. You can enter one or more items of user data separated by commas.	RECEXITINIT	RCXI

Property	Meaning	Long name	Short name
Security exit name	Type either the name of a class that implements the IBM MQ Java interface MQSecurityExit (for a channel security exit written in Java) or a string in the format <i>libraryName(entryPointName)</i> (for a channel security exit not written in Java).	SECEXIT	SCX
Security exit initialization	Type the user data that is passed to the channel security exit when it is called.	SECEXITINIT	SCXI

Broker page

The following table lists the properties that you can set on the **Broker** page of the Connection Factory properties dialog. Edit the properties on the **Broker** page to provide details of the publish/subscribe broker.

Property	Meaning	Long name	Short name
Broker subscription message queue	<p>This property can be used in an IBM WebSphere MQ 7.0, or later, JMS client but has no effect on an IBM WebSphere MQ 7.0, or later, queue manager.</p> <p>Select or type the name of the queue from which non-durable subscription messages are received. The default queue is <code>SYSTEM.JMS.ND.SUBSCRIBER.QUEUE</code>. To use Select to enter the name, you must have already selected a value for the Base Queue Manager property on the Connection page, and the selected queue manager must be running.</p>	BROKERSUBQ	BSUB

Property	Meaning	Long name	Short name
Broker CC subscription message queue	<p>This property can be used in an IBM WebSphere MQ 7.0, or later, JMS client but has no effect on an IBM WebSphere MQ 7.0, or later, queue manager.</p> <p>Select or type the name of the queue from which non-durable subscription messages are retrieved for a ConnectionConsumer. The default queue is <code>SYSTEM.JMS.CC.SUBSCRIBER.QUEUE</code>. To use Select to enter the name, you must have already selected a value for the Base Queue Manager property on the Connection page, and the selected queue manager must be running.</p> <p>For more information, see the BROKERSUBQ property in the IBM MQ online product documentation.</p>	BROKERCCSUBQ	CCSUB
Broker control queue	<p>This property can be used in an IBM WebSphere MQ 7.0, or later, JMS client but has no effect on an IBM WebSphere MQ 7.0, or later, queue manager.</p> <p>Select or type the name of the broker's control queue. To use Select to enter the name, you must have already selected a value for the Broker Queue Manager property on the Connection page, and the selected queue manager must be running.</p>	BROKERCONQ	BCON


Property	Meaning	Long name	Short name
Publication stream	<p>This property can be used in an IBM WebSphere MQ 7.0, or later, JMS client but has no effect on an IBM WebSphere MQ 7.0, or later, queue manager.</p> <p>Select or type the name of the queue where published messages are sent (the stream queue). The default queue is SYSTEM.BROKER.DEFAULT.STREAM. To use Select to enter the name, you must have already selected a value for the Broker Queue Manager property on the Connection page, and the selected queue manager must be running.</p>	BROKERPUBQ	BPUB
Broker version	<p>This property can be used in an IBM WebSphere MQ 7.0, or later, JMS client but has no effect on an IBM WebSphere MQ 7.0, or later, queue manager.</p> <p>Select the version of the broker being used. Click V1 to use an IBM MQ publish/subscribe broker or IBM Integration Bus in compatibility mode; this is the default value if the Transport property is set to Bindings or Client. Click V2 to use IBM Integration Bus in native mode; this is the default value if the Transport property is set to Direct or DirectHTTP.</p>	BROKERVER	BVER

Property	Meaning	Long name	Short name
Publication acknowledgment interval	<p>This property can be used in an IBM WebSphere MQ 7.0, or later, JMS client but has no effect on an IBM WebSphere MQ 7.0, or later, queue manager.</p> <p>Enter the number of messages that are published by a publisher before the IBM MQ JMS client requests an acknowledgment from the broker. If you decrease the value of this property, the client requests acknowledgments more often and, therefore, the performance of the publisher decreases. If you raise the value, the client takes a longer time to throw an exception if the broker fails. This property is relevant only if the Transport property has the value Bindings or Client. The default value is 25.</p>	PUBACKINT	PAI

Temporary queues page

The following table lists the properties that you can set on the **Temporary queues** page of the Connection Factory properties dialog. Edit the properties on the **Temporary queues** page to specify how the JMS temporary queues are dynamically defined.

Property	Meaning	Long name	Short name
Temporary model queue	<p>Select or type the name of the model queue from which JMS temporary queues are created. To use Select to enter the name, you must have already selected a value for the Base Queue Manager property on the Connection page, and the selected queue manager must be running.</p>	TEMPMODEL	TM

Property	Meaning	Long name	Short name
Temporary queue prefix	Type the prefix that is used to form the name of an IBM MQ dynamic queue. The rules for forming the prefix are the same as those for forming the contents of the DynamicQName field in an IBM MQ object descriptor (MQOD) but the last non-blank character must be an asterisk (*). If no value is specified for the property, the value used is AMQ.*.  On z/OS systems, the value used is CSQ.*.	TEMPQPREFIX	TQP

Temporary topics page

The following table lists the properties that you can set on the **Temporary topics** page of the Connection Factory properties dialog. Edit the properties on the **Temporary topics** page to specify how the JMS temporary topics are dynamically defined.

Property	Meaning	Long name	Short name
Temporary topic prefix	Type the prefix that is used to form the name of a temporary topic. When creating temporary topics, JMS generates a topic string of the form TEMP/TEMPTOPICPREFIX/unique_id, or if this property uses the default value, just TEMP/unique_id.	TEMPTOPICPREFIX	TTP

Subscriber page

The following table lists the properties that you can set on the **Subscriber** page of the Connection Factory properties dialog. Edit the properties on the **Subscriber** page to manage subscribers and subscriptions.

Property	Meaning	Long name	Short name
Message selection	<p>This property can be used in an IBM WebSphere MQ 7.0 JMS client but has no effect on an IBM WebSphere MQ 7.0 queue manager.</p> <p>To specify that message selection is done by the JMS client application, click Client. To specify that message selection is done by the broker, click Broker. If the Transport property on the General page has the value Direct, message selection is always done by the broker and the value of the Message Selection property is ignored. Message selection by the broker is not supported when the Broker Version property on the Broker page has the value V1.</p>	MSGSELECTION	MSEL
Sparse subscriptions	<p>This property can be used in an IBM WebSphere MQ 7.0 JMS client but has no effect on an IBM WebSphere MQ 7.0 queue manager.</p> <p>This property controls the message retrieval policy of a TopicSubscriber object. To specify that subscriptions receive frequent matching messages, click No. To specify that subscriptions receive infrequent matching messages, click Yes.</p>	SPARSESUBS	SSUBS

Property	Meaning	Long name	Short name
Subscription store	<p>This property can be used in an IBM WebSphere MQ 7.0 JMS client but has no effect on an IBM WebSphere MQ 7.0 queue manager.</p> <p>Select the location where IBM MQ JMS should store persistent data relating to active subscriptions. To store subscription information on the queues SYSTEM.JMS.ADMIN.QUEUE and SYSTEM.JMS.PS.STATUS.QUEUE, click Queue. To store subscription information in the publish/subscribe broker and not on queues, click Broker. To dynamically select a queue-based or broker-based subscription store depending on the release levels of IBM MQ and the publish/subscribe broker that is installed, click Migrate, which is selected by default. For more information about subscription stores, see Writing IBM MQ classes for JMS applications in the IBM MQ online product documentation.</p>	SUBSTORE	SS
Cleanup level	<p>This property can be used in an IBM WebSphere MQ 7.0 JMS client but has no effect on an IBM WebSphere MQ 7.0 queue manager.</p> <p>Select the cleanup level for subscription stores for which the Subscription Store property on the Broker page is set to Broker or Migrate.</p>	CLEANUP	CL

Property	Meaning	Long name	Short name
Cleanup interval	<p>This property can be used in an IBM WebSphere MQ 7.0 JMS client but has no effect on an IBM WebSphere MQ 7.0 queue manager.</p> <p>Type the number of milliseconds interval between background runs of the publish/subscribe cleanup utility.</p>	CLEANUPINT	CLINT
Clone support	<p>Specify whether two or more instances of the same durable topic subscriber can run simultaneously. Note that enabling clone support contravenes the JMS 1.1 specification. To specify that only one instance of a durable topic subscriber can run at a time, click Disabled. This is the default value. To specify that two or more instances of the same durable topic subscriber can run simultaneously, each instance running in a separate Java virtual machine (JVM), click Enabled.</p>	CLONESUPP	CLS

Property	Meaning	Long name	Short name
Status refresh interval	<p>This property can be used in an IBM WebSphere MQ 7.0 JMS client but has no effect on an IBM WebSphere MQ 7.0 queue manager.</p> <p>Type the number of milliseconds interval between refreshes of the long-running transaction that detects when a subscriber loses its connection to the queue manager. This property is relevant only if the Subscription Store property on the Broker page has the value Queue. For more information about subscription stores, see Writing IBM MQ classes for JMS applications in the IBM MQ online product documentation</p>	STATREFRESHINT	SRI
Wildcard format	<p>Specify which version of wildcard syntax is to be used.</p> <p>Click Only character wildcards to use only character wildcards (for consistency with applications that previously used Broker Version 1; see the Broker version property).</p> <p>Click Only topic wildcards to use only topic level wildcards, which are used in Broker Version 2.</p>	WILDCARDFORMAT	WCFMT

Extended page

The following table lists the properties that you can set on the **Extended** page of the Connection Factory properties dialog. Edit the properties on the **Extended** page to change further properties of the connection factory object.

Property	Meaning	Long name	Short name
Message batch size	<p>This property can be used in an IBM WebSphere MQ 7.0 JMS client but has no effect on an IBM WebSphere MQ 7.0 queue manager.</p> <p>Type the maximum number of messages to be taken from a queue in one packet when using asynchronous message delivery. The default is 10.</p>	MSGBATCHSZ	MBS
Message retention	<p>Specify whether the connection consumer keeps unwanted messages on the input queue. To keep unwanted messages on the input queue, click Yes. To deal with unwanted messages according to their disposition options, click No.</p>	MSGRETENTION	MRET
Polling interval	<p>This property can be used in an IBM WebSphere MQ 7.0 JMS client but has no effect on an IBM WebSphere MQ 7.0 queue manager.</p> <p>Type the number of milliseconds interval that elapses before each message listener tries again to get a message from its queue when each message listener within a session has no suitable message on its queue. The default is 5000. If there is frequently no suitable message available for any of the message listeners in the session, consider increasing the value of the property.</p>	POLLINGINT	PINT

Property	Meaning	Long name	Short name
Rescan interval	<p>This property can be used in an IBM WebSphere MQ 7.0 JMS client but has no effect on an IBM WebSphere MQ 7.0 queue manager.</p> <p>Type the number of milliseconds interval before the IBM MQ JMS client returns to the beginning of the queue when searching for a suitable message. When a message consumer in the point-to-point domain uses a message selector to select which messages it wants to receive, the IBM MQ JMS client searches the IBM MQ queue for suitable messages in the sequence determined by the Message Delivery Sequence property of the queue (see Queue properties). When the client finds a suitable message and delivers it to the consumer, the client resumes the search for the next suitable message from its current position in the queue. The client continues to search the queue in this way until it reaches the end of the queue, or until the interval of time in milliseconds, as determined by the value of this property, has expired. In each case, the client returns to the beginning of the queue to continue its search, and a new time interval commences.</p>	RESCANINT	RINT

Property	Meaning	Long name	Short name
Fail if quiesce	To specify that calls to certain methods fail if the queue manager is in a quiescing state, click Yes . If an application detects that the queue manager is quiescing, the application can complete its immediate task and close the connection, allowing the queue manager to stop. To specify that no method call fails because the queue manager is in a quiescing state, click No . If you click No , an application cannot detect that the queue manager is quiescing so the application might continue to perform operations against the queue manager and, therefore, prevent the queue manager from stopping.	FAILIFQUIESCE	FIQ
Syncpoint all gets	To specify that all gets should be performed under syncpoint, click Yes . To specify that all gets should not be performed under syncpoint, click No .	SYNCPOINTALLGETS	SPAG

Property	Meaning	Long name	Short name
Target client matching	Specify whether a reply message, sent to the queue identified by the JMSReplyTo header field of an incoming message, has an MQRFH2 header only if the incoming message has an MQRFH2 header. To specify that if an incoming message does not have an MQRFH2 header, the Target Client property on the destination queue is set to MQ, click Yes . If the incoming message does have an MQRFH2 header, the Target Client property is set to JMS instead. To specify that the Target Client property of the destination queue is always set to JMS, click No .	TARGCLIENTMATCHING	TCM
Asynchronous error check interval	Type the number of send calls to allow between checking for asynchronous put errors, within a single non-transacted JMS session. The minimum value is 0; the value can be any positive integer.	SENDCHECKCOUNT	SCC

Advanced tuning page

The following table lists the properties that you can set on the **Advanced tuning** page of the Connection Factory properties dialog. Edit the properties on the **Advanced tuning** page to configure advanced settings. For most systems, the default settings are appropriate.

Property	Meaning	Long name	Short name
Process duration	<p>This property can be used in an IBM WebSphere MQ 7.0 JMS client but has no effect on an IBM WebSphere MQ 7.0 queue manager.</p> <p>To specify that a subscriber can give no guarantee about how quickly it can process any message it receives, click Unknown (the default value). To specify that a subscriber guarantees to process any message it receives before returning control to the IBM MQ JMS client, click Short.</p>	PROCESSDURATION	PROCDUR
Optimistic publication	<p>This property can be used in an IBM WebSphere MQ 7.0 JMS client but has no effect on an IBM WebSphere MQ 7.0 queue manager.</p> <p>To specify that when a publisher publishes a message, the IBM MQ JMS client does not return control to the publisher until it has completed all the processing associated with the call and can report the outcome to the publisher, click No (the default value). To specify that when a publisher publishes a message, the IBM MQ JMS client returns control to the publisher immediately, before it has completed all the processing associated with the call and can report the outcome to the publisher, click Yes (the IBM MQ JMS client reports the outcome only when the publisher commits the message).</p>	OPTIMISTICPUBLICATION	OPTPUB

Property	Meaning	Long name	Short name
Outcome notification	<p>This property can be used in an IBM WebSphere MQ 7.0 JMS client but has no effect on an IBM WebSphere MQ 7.0 queue manager.</p> <p>To specify that when a subscriber acknowledges or commits a message, the IBM MQ JMS client does not return control to the subscriber until it has completed all the processing associated with the call and can report the outcome to the subscriber, click Yes (the default value). To specify that when a subscriber acknowledges or commits a message, the IBM MQ JMS client returns control to the subscriber immediately, before it has completed the call and can report the outcome to the subscriber, click No.</p>	OUTCOMENOTIFICATION	NOTIFY
Receive isolation	<p>This property can be used in an IBM WebSphere MQ 7.0 JMS client but has no effect on an IBM WebSphere MQ 7.0 queue manager.</p> <p>To specify that a subscriber receives only the messages on the subscriber queue that have been committed, click Committed (the default value). To specify that a subscriber can receive messages that have not been committed on the subscriber queue, click Uncommitted. The value of Uncommitted has an effect only if the Process Duration property has the value Short.</p>	RECEIVEISOLATION	RCVISOL

Related reference

[“Strings in property dialogs” on page 655](#)

To include certain characters in a string, you must punctuate the string in a particular way.

Destination properties

You can view and set destination properties in the Destination properties dialog. The properties that are available in the dialog depend on the type of destination.

The following tables list all the properties that you can set for destinations:

- [General](#)
- [Message handling](#)
- [Broker](#)
- [Producers](#)
- [Consumers](#)
- [Extended](#)

For each property, there is a brief description of when you might need to configure the property. The tables also give the equivalent long and short names to use in the JMS Administration command-line tool. The properties that are available in the Properties dialog depend on the type of destination; queue destinations have some different properties from topic destinations. For more information, see [Configuring JMS objects using the administration tool](#) in the online IBM MQ product documentation.

General page

The following table lists the properties that you can set on the **General** page of the Destination properties dialog.

Table 6. . General page properties			
Property	Meaning	Long name	Short name
Name	This shows the name of the object.	NAME	
Description	Type a description of the object.	DESCRIPTION	DESC
Class name	This shows the class name that is implemented by the destination.		
Messaging provider	This shows the transport that is supported by the destination object and can be either IBM MQ or Real-time transport. Real-time transport is not available in IBM MQ 8.0. If you are using IBM MQ 8.0 you can define Real-time transport, but it fails when an attempt is made to create a connection.		
Queue manager	Select or type the name of the queue manager that hosts the destination queue.	QMANAGER	QMGR

Table 6. . General page properties (continued)			
Property	Meaning	Long name	Short name
Queue	Select the name of the queue that is represented by the destination.	QUEUE	QU
Topic	Type the name of the topic that this destination represents.	TOPIC	TOP

Message handling page

The following table lists the properties that you can set on the **Message handling** page of the Destination properties dialog. Edit the properties on the **Message handling** page to configure what happens to messages that are sent to the destination.

Table 7. . Message handling page properties			
Property	Meaning	Long name	Short name
Expiry	Specify the period after which messages at the destination expire. To specify that expiry can be defined by the JMS client application, click Application . To specify that no expiry occurs, click Unlimited . Otherwise, type the number of milliseconds before the messages expire.	EXPIRY	EXP

Table 7. . Message handling page properties (continued)

Property	Meaning	Long name	Short name
Persistence	Specify the persistence of messages that are sent to the destination. To specify that the persistence is defined by the JMS application, click Application . To specify that the persistence takes the value of the queue's default value, click Queue default . To specify that messages are persistent, click Persistent . To specify that messages are not persistent, click Not persistent . To specify that nonpersistent messages on the queue are not discarded when the queue manager restarts following a quiesced or immediate shutdown, click High (nonpersistent messages might be discarded, however, following a preemptive shutdown or a failure).	PERSISTENCE	PER
Priority	Specify the priority for messages that are sent to the destination. To specify that the priority is defined by the JMS client application, click Application . To specify that the priority takes the value of the queue default, click Queue default . Otherwise, type the priority, from 0 to 9.	PRIORITY	PRI

Table 7. . Message handling page properties (continued)

Property	Meaning	Long name	Short name
MQMD Message Context	<p>Specify the context when sending messages to a destination. There are three options to choose from:</p> <p>Default: The MQOPEN API call and the MQPMO structure specifies no explicit message context options. This is the default value.</p> <p>Set All Context: The MQOPEN API call specifies the message context option MQOO_SET_ALL_CONTEXT and the MQPMO structure specifies MQPMO_SET_ALL_CONTEXT.</p> <p>Set Identity Context: The MQOPEN API call specifies the message context option MQOO_SET_IDENTITY_CONTEXT and the MQPMO structure specifies MQPMO_SET_IDENTITY_CONTEXT.</p>	MDMSGCTX	MDCTX
MQMD Write Enabled	<p>Specify whether a JMS application can set the values of MQMD fields. There are two options to choose from:</p> <p>No: All JMS_IBM_MQMD* properties are ignored and their values are not copied into the underlying MQMD structure. This is the default value.</p> <p>Yes: All JMS_IBM_MQMD* properties are processed. Their values are copied into the underlying MQMD structure.</p>	MDWRITE	MDW

Table 7. . Message handling page properties (continued)

Property	Meaning	Long name	Short name
MQMD Read Enabled	<p>Specify whether a JMS application can extract the values of MQMD fields. There are two options to choose from:</p> <p>No: When sending messages, the JMS_IBM_MQMD* properties on a sent message are not updated to reflect the updated field values in the MQMD. When receiving messages, none of the JMS_IBM_MQMD* properties are available on a received message, even if the sender has set some or all of them. This is the default value.</p> <p>Yes: When sending messages, all of the JMS_IBM_MQMD* properties on a sent message are updated to reflect the updated field values in the MQMD, including those properties that the sender did not explicitly set. When receiving messages, all of the JMS_IBM_MQMD* properties are available on a received message, including those properties that the sender did not explicitly set.</p>	MDREAD	MDR

Table 7. . Message handling page properties (continued)

Property	Meaning	Long name	Short name
Message Body	<p>Specify whether a JMS application processes the MQRFH2 of an IBM MQ message as part of the JMS message body. There are three options to choose from:</p> <p>Unspecified: When sending messages; IBM MQ classes for JMS does or does not generate and include an MQRFH2 header, depending on the value of WMQ_TARGET_CLIENT. When receiving messages, it acts as if the value is set to JMS. This is the default value.</p> <p>JMS: When sending messages; IBM MQ classes for JMS automatically generates an MQRFH2 header and includes it in the IBM MQ message. When receiving messages; IBM MQ classes for JMS set the JMS message properties according to values in the MQRFH2 (if present); it does not present the MQRFH2 as part of the JMS message body.</p> <p>MQ: When sending messages; IBM MQ classes for JMS does not generate an MQRFH2. When receiving messages; IBM MQ classes for JMS presents the MQRFH2 as part of the JMS message body.</p>	MSGBODY	MBODY

Table 7. . Message handling page properties (continued)

Property	Meaning	Long name	Short name
Receive conversion	Configures use of MQGMO_CONVERT option. When requesting messages, selects if conversion is to be carried out in the queue manager by specifying MQGMO_CONVERT on the MQGET, or in the client application.	RECEIVECONVERSION	RCNV
Receive CCSID	Requested CCSID to convert the message to. Used when the IBM MQ classes for JMS are requesting the queue manager to perform conversion (for example by specifying WMQ_RECEIVE_CONVERSION_QMGR as the argument to setReceiveConversion). The value of this property is the CCSID which the queue manager is requested to convert the message to. The default value is CCSID 1208.	RECEIVECCSID	RCCS

Broker page

The following table lists the properties that you can set on the **Broker** page of the Destination properties dialog. Edit the properties on the **Broker** page to provide details of the publish/subscribe broker.

Table 8. . Broker page properties

Property	Meaning	Long name	Short name
Broker durable subscription queue	<p>This property can be used in an IBM WebSphere MQ 7.0, or later, JMS client but has no effect on an IBM WebSphere MQ 7.0, or later, queue manager.</p> <p>If the queue manager that the JMS client is connecting to is the same one that is used by the publish subscribe broker, you can select a queue from the broker queue manager. The selected queue manager must be running.</p> <p>If this is not the case, then you must type the name of the queue from which durable subscription messages are retrieved for. The default queue is SYSTEM.JMS.D.SUBSCRIBER.QUEUE.</p>	BROKERDURSUBQ	BDSUB
Broker CC durable subscription message queue	<p>This property can be used in an IBM WebSphere MQ 7.0, or later, JMS client but has no effect on an IBM WebSphere MQ 7.0, or later, queue manager</p> <p>If the queue manager that the JMS client is connecting to is the same one that is used by the publish subscribe broker, you can select a queue from the broker queue manager. The selected queue manager must be running.</p> <p>If this is not the case, then you must type the name of the queue from which durable subscription messages are retrieved for a ConnectionConsumer. The default queue is SYSTEM.JMS.D.CC.SUBSCRIBER.QUEUE.</p>	BROKERCCDURSUBQ	CCDSUB

Table 8. . Broker page properties (continued)

Property	Meaning	Long name	Short name
Publication stream	<p>This property can be used in an IBM WebSphere MQ 7.0, or later, JMS client but has no effect on an IBM WebSphere MQ 7.0, or later, queue manager.</p> <p>Select or type the name of the queue to which published messages are sent (the stream queue). The default queue is SYSTEM.BROKER.DEFAULT.STREAM. To use Select to enter the name, you must have already selected a value for the Broker Publication Queue Manager property, and the selected queue manager must be running.</p>	BROKERPUBQ	BPUB
Broker publication queue manager	Select or type the name of the queue manager that owns the queue to which messages published on the topic are sent.	BROKERPUBQMGR	BPQM
Broker version	<p>This property can be used in a IBM WebSphere MQ 7.0, or later, JMS client but has no effect on an IBM WebSphere MQ 7.0, or later, queue manager.</p> <p>Select the version of the broker being used. Click V1 to use an IBM MQ publish/subscribe broker, or IBM Integration Bus in compatibility mode; this is the default value if the Transport property is set to Bindings or Client. Click V2 to use a broker of IBM Integration Bus in native mode; this is the default value if the Transport property is set to Direct or DirectHTTP.</p>	BROKERVER	BVER

Producers page

The following table lists the properties that you can set on the **Producers** page of the Destination properties dialog. Edit the properties on the **Producers** page to change further properties of the destination object.

Table 9. . Producers page properties			
Property	Meaning	Long name	Short name
Allow asynchronous puts	Specify whether message producers are allowed to use asynchronous puts to send messages to this destination. To determine whether message producers are allowed by referring to the queue or topic definition, click As destination (default). To allow asynchronous puts, click Enabled ; to disallow asynchronous puts, click Disabled .	PUTASYNCAALLOWED	PAA

Consumers page

The following table lists the properties that you can set on the **Consumers** page of the Destination properties dialog. Edit the properties on the **Consumers** page to change further properties of the destination object.

Table 10. . Consumers page properties			
Property	Meaning	Long name	Short name
Allow read ahead	Specify whether message consumers and queue browsers are allowed to use read ahead to get non-persistent messages from this destination into a client buffer before receiving them. To determine whether read ahead is allowed by referring to the queue or topic definition, click As destination (default). To allow read ahead, click Enabled ; to disallow read ahead, click Disabled .	READAHEADALLOWED	RAA [®]

Table 10. . Consumers page properties (continued)

Property	Meaning	Long name	Short name
Read ahead close policy	For messages being delivered to an asynchronous message listener, specify what should happen to messages on the client proxy queue when the message consumer is closed. To specify that all messages on the client proxy queue are delivered to the application's MessageListener before returning, click Deliver all (default). To specify that the current MessageListener invocation only completes before returning, potentially leaving further messages on the client proxy queue to be discarded, click Deliver current .	READAHEADCLOSEPOLICY	RACP
Wildcard format	Specify which version of wildcard syntax is to be used. Click Only character wildcards to use only character wildcards (for consistency with applications that previously used Broker Version 1; see the Broker version property). Click Only topic wildcards to use only topic level wildcards, which are used in Broker Version 2.	WILDCARDFORMAT	WCFMT

Extended page

The following table lists the properties that you can set on the **Extended** page of the Destination properties dialog. Edit the properties on the **Extended** page to change further properties of the destination object.

Table 11. . Extended page properties

Property	Meaning	Long name	Short name
Coded character set ID	Type the coded character set ID (CCSID) to be used for destinations.	CCSID	CCS
Encoding	Select the encoding scheme used for this destination. For more information, see Configuring JMS objects using the administration tool in the IBM MQ online product documentation.	ENCODING	ENC
Fail if quiesce	Specify whether calls to certain methods fail if the queue manager is in a quiescing state. To specify that calls to certain methods fail if the queue manager is in a quiescing state, click Yes . This means that if an application detects that the queue manager is quiescing, the application can complete its immediate task and close the connection, allowing the queue manager to stop. To specify that no method calls fail because the queue manager is in a quiescing state, click No . This means that an application cannot detect that the queue manager is quiescing, so the application might continue to perform operations against the queue manager and, therefore, prevent the queue manager from stopping.	FAILIFQUIESCE	FIQ

Table 11. . Extended page properties (continued)

Property	Meaning	Long name	Short name
Multicast	<p>Specify whether messages are delivered to message consumers that are using multicast transport. Multicast transport is applicable only to topic destinations and can be used only when the connection factory uses Direct IP transport. Direct IP transport is not available in IBM MQ 8.0. If you are using IBM MQ 8.0 you can define Direct IP transport, but it fails when an attempt is made to create a connection.</p> <p>The default value is As Connection Factory, which means that the value of the connection factory's Multicast property is used.</p> <p>Click Enabled to deliver messages to the message consumer that is using multicast transport. The topic must be configured for reliable multicast in the broker; a reliable quality of service is used if the topic is configured for reliable multicast.</p> <p>Click Reliable to deliver messages to the message consumer that is using multicast transport with a reliable quality of service. The topic must be configured for reliable multicast, in the broker; if the topic is not configured for reliable multicast, you cannot create a message consumer for the topic.</p> <p>Click Not reliable to deliver messages to the message consumer that is using multicast transport but without using a reliable quality of service. The topic</p>	MULTICAST	MCAST

Table 11. . Extended page properties (continued)

Property	Meaning	Long name	Short name
Target client	<p>Click JMS to specify that the target of the message is a JMS application.</p> <p>Click MQ to specify that the target of the message is a non-JMS IBM MQ application.</p>	TARGCLIENT	TC
ReplyTo destination style	<p>Specify the format of the JMSReplyTo field.</p> <p>Click Default to use the default value. The default value is equivalent to the information in the RFH2 unless the JVM system property has been set.</p> <p>Click MQMD to use the value supplied in the MQMD. This will populate the queue manager field with the value from the MQMD.</p> <p>Click RFH2 to use the value supplied in the RFH2 header. If the sending application set a JMSReplyTo value then that value will be used.</p>	REPLYTOSTYLE	RTOST
Receive conversion	<p>Configures use of MQGMO_CONVERT option . When requesting messages, selects if conversion is to be carried out in the queue manager by specifying MQGMO_CONVERT on the MQGET, or in the client application.</p>	RECEIVECONVERSION	RCNV

Table 11. . Extended page properties (continued)			
Property	Meaning	Long name	Short name
Receive CCSID	Requested CCSID to convert the message to . Used when the IBM MQ classes for JMS are requesting the queue manager to perform conversion (for example by specifying WMQ_RECEIVE_CONVERSION_QMGR as the argument to setReceiveConversion). The value of this property is the CCSID which the queue manager is requested to convert the message to. The default value is CCSID 1208.	RECEIVECCSID	RCCS

Related concepts

[“JMS destinations \(queues and topics\)” on page 208](#)

A JMS destination is an object (a JMS queue or a JMS topic) that represents the target of messages that the client produces and the source of messages that the client consumes. In point-to-point messaging, destinations represent queues; in publish/subscribe messaging, destinations represent topics.

[“Topics” on page 30](#)

A topic identifies what a publication is about. A topic is a character string that describes the subject of the information that is published in a Publish/Subscribe message. As a subscriber, you can specify a topic or range of topics by using wildcards to receive the information that you require.

Related reference

[“Strings in property dialogs” on page 655](#)

To include certain characters in a string, you must punctuate the string in a particular way.

[“IBM MQ queue properties” on page 412](#)

The attributes that you can set for a queue depend on the type of queue. Different types of IBM MQ queues have different properties. Some of the attributes do not apply to all types of queue, some attributes are specific to cluster queues, and some attributes are specific to z/OS queues.

[“Topic properties” on page 458](#)


An IBM MQ topic is an IBM MQ object that identifies what a publication is about. You can set attributes for topics. Some topic attributes are specific to z/OS topics. Also, there are some attributes that you can alter only while you are creating a topic. You cannot modify these attributes after the IBM MQ topic has been created.

Status attributes

In IBM MQ Explorer, you can view the current status of IBM MQ objects. For example, you can find out whether a channel is running, or you can find out when the last message was put on a certain queue. You can also view the saved status of a channel.

The following topics list all of the status attributes for IBM MQ objects. For each attribute, there is a description of what the attribute shows.

- [Queue managers](#)
- [Queue manager Pub/Sub Engines](#)


- [Queues](#)
- [Topics](#)
- [Subscriptions](#)
- [Topic subscribers](#)
- [Topic publishers](#)
- [Channels](#)
- [Listeners](#)
- [Custom services](#)
-  [Coupling facility](#)
- [“Display SMDS status attributes” on page 654](#)

Related tasks

[“Viewing the status of objects” on page 202](#)

You can display the current status of any object that can be in different states, in IBM MQ Explorer. For IBM MQ channels, you can also view the saved status.















Queue manager status attributes

The status attributes of distributed queue managers , and z/OS queue managers.

Distributed queue manager status

This table lists the status attributes of distributed queue managers. For each attribute, there is a brief description of what the attribute shows. The table also gives the equivalent MQSC parameter for the DISPLAY QMSTATUS command. For more information about MQSC commands, see [Script \(MQSC\) Commands](#) in the IBM MQ online product documentation.

Attribute	Meaning	MQSC parameter
Queue manager name	The name of the queue manager.	
Queue manager status	The status of the queue manager, which can be: Starting, Running, or Quiescing.	STATUS
Connection count	The current number of connections to the queue manager.	CONNS
Channel initiator status	The status of the channel initiator, which can be: Stopped, Starting, Running, or Stopping.	CHINIT
Command server status	The status of the command server, which can be: Stopped, Starting, Running, or Stopping.	CMDSERV
Installation description	Description of the installation associated with the queue manager.	
Installation name	Name of the installation associated with the queue manager.	
Installation path	Path of the installation associated with the queue manager.	

Attribute	Meaning	MQSC parameter
  Archive log extent name	Name of the oldest log extent for which the queue manager is waiting for archive notification.	ARCHLOG
  Archive size	The amount of space occupied, in megabytes, by log extents no longer required for restart or media recovery, but waiting to be archived.	ARCHSZ
Current log extent name	The name of the log extent that is being written to at the time that you open the status dialog.	CURRLOG
  Log in use	The percentage of the primary log space in use for restart recovery at this point in time.	LOGINUSE
  Log utilization	A percentage estimate of how well the queue manager workload is contained within the primary log space.	LOGUTIL
Restart recovery log extent name	The name of the oldest log extent required by the queue manager to perform restart recovery.	RECLOG
Media recovery log extent name	The name of the oldest log extent required by the queue manager to perform media recovery.	MEDIALOG
  Media size	Size of the log data required for media recovery in megabytes.	MEDIASZ
  Restart recovery size	Size of the log data required for restart recovery in megabytes.	RECSZ
  Reuse size	The amount of space occupied, in megabytes, by log extents available to be reused.	REUSESZ
Log path	<p>The path of the queue manager event log. The path is of this format:</p> <pre><i>installationlocation</i>\WebSphere MQ\log\<i>queuemanager</i>\active\</pre> <p>where <i>installationlocation</i> is the location where IBM MQ is installed, and <i>queuemanager</i> is the name of the queue manager.</p>	LOGPATH
Start date	The date on which the queue manager was started.	STARTDA
Start time	The time at which the queue manager was started.	STARTTI

z/OS queue manager status

This section lists the status attributes of z/OS queue managers.

Channel initiator

This table lists the channel initiator status attributes of z/OS queue managers. For each attribute, there is a brief description of what the attribute shows. The equivalent MQSC command is DISPLAY CHINIT. For more information about MQSC commands, see [Script \(MQSC\) Commands](#) in the IBM MQ online product documentation.

Attribute	Meaning
TCP name	The TCP system name.
Active adapters	The number of active adapter subtasks.
Active adapters requested	The requested number of adapter subtasks.
Active dispatchers	The number of active dispatchers.
Active dispatchers requested	The requested number of dispatchers.
Active SSL tasks	The number of active TLS server subtasks.
Active SSL tasks requested	The requested number of TLS server subtasks.
Current [®] channel connections	The number of current channel connections.
Channel connections requested	The requested number of channel connections.
Current TCP/IP channels	The number of current TCP/IP channel connections.
Current LU6.2 channels	The number of current LU6.2 channel connections.
Active channel connections	The number of active channel connections.
Active channel connections requested	The requested number of active channel connections
Active channels paused	The number of active channel connections that have been paused, waiting to become active, because the limit for active channels has been reached.
Active channels started	The number of active channel connections that have started.
Active channels stopped	The number of active channel connections that have stopped, requiring manual intervention.
Active channels retrying	The number of active channel connections that are attempting to reconnect following a temporary error.
Channel initiator status	The status of the channel initiator, which can be: Stopped, Starting, Running, or Stopping.

Log

This table lists the log status attributes of z/OS queue managers. For each attribute, there is a brief description of what the attribute shows. The equivalent MQSC command is DISPLAY LOG. For more information about MQSC commands, see [Script \(MQSC\) Commands](#) in the IBM MQ online product documentation.

Attribute	Meaning
Parameter type	This attribute shows which type of information is displayed in the table.

Attribute	Meaning
Queue manager start time	The time that the queue manager was started.
Queue manager start date	The date on which the queue manager was started.
RBA of queue manager start	The Relative Byte Address (RBA) from which logging began when the queue manager was started.
RBA of log	The Relative Byte Address (RBA) of the most recently written log record.
Logging suspended	Specifies whether logging is suspended.
Status of the offload task	The status of the offload task. <code>Alloc archive</code> means that the offload task is busy allocating the archive data set, which could indicate that a tape mount request is pending; <code>Copying BSDS</code> means that the offload task is busy copying the BSDS data set; <code>Busy</code> means that the offload task is busy with other processing; <code>Available</code> means that the offload task is waiting for work.
Active logs	The number of active log data sets.
Full active logs	The total number of full active log data sets that have not yet been archived.

Usage

The following tables list the Usage status attributes of z/OS queue managers. For each attribute, there is a brief description of what information the attribute shows. The table also gives the equivalent MQSC parameter for the `DISPLAY USAGE` command, which displays information about the current state of a page set, or displays information about the log data sets. For more information about MQSC commands, see [Script \(MQSC\) Commands](#) in the IBM MQ online product documentation.

Dataset records

This table shows the attributes that are displayed in the **Dataset records** table.

Attribute	Meaning
Usage type	This attribute shows which type of information is displayed in the table.
RBA of log	The Relative Byte Address (RBA) of the most recently written log record.
Data set type	The type of data set, and circumstance. <code>Oldest Active UOW</code> means that the log data set contains the start RBA of the oldest active unit of work for the queue manager; <code>Oldest pageset recovery</code> means that the log data set contains the oldest restart RBA of any page set for the queue manager; <code>Oldest CF struc recovery</code> means that the log data set contains the LRSN which matches the time of the oldest current backup of any CF structure in the queue sharing group.
LRSN of log	The Log Record Sequence Number (LRSN) of the most recently written log record.

Attribute	Meaning
Data set name	The name of the data set.

Buffer Pool records

This table shows the attributes that are displayed in the **Buffer Pool records** table.

Attribute	Meaning
Usage type	This attribute shows which type of information is displayed in the table.
Buffer pool ID	The buffer pool identifier, which identifies the buffer pool being used by the page set.
Buffers defined	The number of buffers defined for the buffer pool.
Page class	The type of virtual storage pages used for backing the buffers in the buffer pool. The values for page class are: <ul style="list-style-type: none"> • Pageable 4 KB pages • Fixed 4 KB pages
Buffer pool location	Information about the LOCATION value for individual buffer pools. The values for LOCATION are: <ul style="list-style-type: none"> • Above the bar (64 bit storage) • Below the bar (31 bit storage) • Switching to above the bar (64 bit storage) • Switching to below the bar (31 bit storage)
Free buffers	The number of unused buffers in the buffer pool.
Free buffers (%)	The percentage of unused buffers in the buffer pool.

Pageset records

This table shows the attributes that are displayed in the **Pageset records** table.

Attribute	Meaning
Usage type	This attribute shows which type of information is displayed in the table.

Attribute	Meaning
Expand type	<p>How the queue manager expands a page set when it becomes nearly full, and further pages are required within it. None means that no further page set expansion is to take place. User means that the secondary extent size that was specified when the page set was defined is used. If no secondary extent size was specified, or it was specified as zero, then no dynamic page set expansion can take place. System means that the secondary extent size that was specified when the page set was defined is ignored. Instead, the queue manager sets a value that is approximately 10 per cent of the current size of the page set. The new extent size is rounded up to the nearest cylinder of DASD. If no secondary extend size was specified, or was specified as zero, dynamic page set expansion can still take place. The queue manager sets a value that is approximately 10 per cent of the current size of the page set. The new extent size is rounded up depending on the characteristics of the DASD. If the value is User or System, page set expansion occurs when the space in the page is 90 per cent used, and is performed asynchronously to any other page set activity. At restart, if a previously used page set has been replaced with a data set that is smaller, it is expanded until it reaches the size of the previously used data set. Only one extent is required to reach this size.</p>
Page set ID	<p>The page set identifier. This is an optional number from 00 to 99. An asterisk on its own specifies all page set identifiers.</p>
Pages	<p>The total number of 4 KB pages in the page set.</p>
Unused pages	<p>The number of pages that are not use (that is, available page sets).</p>
Pages holding persistent data	<p>The number of pages holding persistent data. These pages are being used to store object definitions and persistent message data.</p>
Pages holding nonpersistent data	<p>The number of pages holding nonpersistent data. These pages are being used to store nonpersistent message data.</p>
Extents used at restart	<p>The number of extents used by the page set at restart.</p>
Expand count	<p>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 123, provided that enough space is available.</p>

Attribute	Meaning
Page set status	The current status of the page set. Available means that the page set is available; Defined means that the page set has been defined but has never been used; Offline means that 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; Not defined means that the page set is not defined to the queue manager.
Buffer pool ID	The buffer pool identifier, which identifies the buffer pool being used by the page set.

Shared message data set records

This table shows the attributes that are displayed in the **Shared message data set records** table.

Attribute	Meaning
Status	The status of the shared message data set records for the selected queue manager.
Application structure	This is the name of the application structure for the selected queue manager.
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 total 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 percentage of used data blocks to the total data blocks.
Block size (KB)	This shows the size of each buffer in KB. This is equal to the logical block size of the shared message data set.
Total buffers	This is the number of buffers in the pool
In use buffers	This is the number of buffers which are currently being used by requests to transfer data to or from the data set.
Saved buffers	This is the number of buffers which are free but currently contain saved data for recently accessed blocks.

Attribute	Meaning
Empty buffers	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 current 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.
Wait rate (%)	This is the percentage of requests to acquire a buffer which had to wait for a free buffer.

Related concepts

[“Queue managers” on page 28](#)

A queue manager is a program that provides messaging services to applications. Applications that use the Message Queue Interface (MQI) can put messages on queues and get messages from queues. The queue manager ensures that messages are sent to the correct queue or are routed to another queue manager.

Related tasks

[“Viewing the status of objects” on page 202](#)

You can display the current status of any object that can be in different states, in IBM MQ Explorer. For IBM MQ channels, you can also view the saved status.

Related reference

[“Queue manager Publish/Subscribe Engine status attributes” on page 612](#)

The status attributes of the queue manager Publish/Subscribe Engine.

Queue manager Publish/Subscribe Engine status attributes

The status attributes of the queue manager Publish/Subscribe Engine.

For each attribute, there is a brief description of what the attribute shows. The table also gives the equivalent MQSC parameter. For more information about MQSC commands, see [Script \(MQSC\) Commands](#) in the IBM MQ online product documentation.

This table lists the status attributes in the Local section:		
Attribute	Meaning	MQSC parameter
Queue manager name	The name of the local queue manager.	QMNAME

This table lists the status attributes in the Local section: <i>(continued)</i>		
Attribute	Meaning	MQSC parameter
Status	<p>The current status of the Publish/Subscribe Engine on the local queue manager. The status value can be:</p> <p>Active which means the Publish/Subscribe Engine and the queued Publish/Subscribe interface are running. It is possible to Publish/Subscribe via the MQI and the queues being monitored by the queued Publish/Subscribe interface.</p> <p>Starting which means the Publish/Subscribe Engine is initializing and is not yet operational.</p> <p>Stopping which means the Publish/Subscribe Engine is stopping.</p> <p>Compatability The Publish/Subscribe Engine is running but the queued Pub/Sub interface is not active. Any messages put to the queues monitored by the queued Pub/Sub interface will not be processed.</p> <p>Error which means there is an error with the connection from the Publish/Subscribe Engine to the local queue manager. The error logs will contain more information about the error.</p> <p>Inactive which means the Publish/Subscribe Engine is not active.</p>	STATUS
Sub count	Shows the total number of subscriptions against the local topic tree.	SUBCOUNT
Topic count	Shows the total number of topic nodes in the local topic tree. If the number is growing it might indicate that a shorter tree life is required, or that a redesign of the topics themselves is required.	TPCOUNT

This table lists the status attributes in the Parent section:		
Attribute	Meaning	MQSC parameter
Queue manager name	The name of the parent queue manager.	QMNAME
Status	<p>The current status of the Publish/Subscribe Engine on the parent queue manager. The status value can be:</p> <p>Active which means the connection with the parent queue manager is active.</p> <p>Starting which means the queue manager is attempting to request that another queue manager become its parent.</p> <p>Stopping which means the queue manager is disconnecting from its parent.</p> <p>Refused which means the connection has been refused by the parent queue manager. This might be caused by the parent queue manager already having another child queue manager of the same name.</p> <p>Error which means there is an error with the connection from the Publish/Subscribe Engine to the parent queue manager. Possible reasons for the error are: The transmit queue is not defined, or the transmit queue put is disabled.</p>	STATUS
Sub count	Queue manager relations are not inquired for parents, and no value is returned.	SUBCOUNT
Topic count	Queue manager relations are not inquired for parents, and no value is returned.	TPCOUNT

This table lists the status attributes in the Children section:		
Attribute	Meaning	MQSC parameter
Queue manager name	The name of the child queue manager.	QMNAME

This table lists the status attributes in the Children section: <i>(continued)</i>		
Attribute	Meaning	MQSC parameter
Status	<p>The current status of the Publish/Subscribe Engine on the child queue manager. The status value can be:</p> <p>Active which means the connection with the child queue manager is active.</p> <p>Starting which means that another queue manager is attempting to request that this queue manager become its parent.</p> <p>Stopping which means the child queue manager is disconnecting.</p> <p>Error which means there is an error with the connection from the Publish/Subscribe Engine to the child queue manager. Possible reasons for the error are: The transmit queue is not defined, or the transmit queue put is disabled.</p>	STATUS
Sub count	Queue manager relations are not inquired for children, and no value is returned.	SUBCOUNT
Topic count	Queue manager relations are not inquired for children, and no value is returned.	TPCOUNT

Related concepts

[“Queue managers” on page 28](#)

A queue manager is a program that provides messaging services to applications. Applications that use the Message Queue Interface (MQI) can put messages on queues and get messages from queues. The queue manager ensures that messages are sent to the correct queue or are routed to another queue manager.

Related tasks

[“Viewing the status of objects” on page 202](#)

You can display the current status of any object that can be in different states, in IBM MQ Explorer. For IBM MQ channels, you can also view the saved status.

Related reference


[“Queue manager status attributes” on page 605](#)



The status attributes of distributed queue managers , and z/OS queue managers.

Queue status attributes

The status attributes of queues and handles that are accessing the queues.

For each attribute, there is a brief description of what information the attribute shows. The table also gives the equivalent MQSC parameter for the DISPLAY QSTATUS command. For more information about MQSC commands, see [Script \(MQSC\) Commands](#) in the IBM MQ online product documentation.

Attribute	Meaning	MQSC parameter
Queue name	The name of the queue.	
Current queue depth	The number of messages currently on the queue.	CURDEPTH
Open input count	This is the number of applications that are currently connected to the queue to get messages from the queue.	IPPROCS
Open output count	This is the number of applications that are currently connected to the queue to put messages on the queue.	OPPROCS
Uncommitted messages	This indicates whether there are any uncommitted changes (puts and gets) pending for the queue. If there are uncommitted changes pending, the value is a number corresponding to the number of uncommitted messages that there are pending (1, 2, 3, 4, 5, etc). If there are no uncommitted changes pending, the value is No.  For z/OS shared queues, the value applies only to the queue manager that is generating the reply. The value does not apply to all of the queue managers in the queue sharing group.	UNCOM
Media recovery log extent name	The name of the oldest log extent required by the queue to perform media recovery.	MEDIALOG
Queue monitoring	The log extent or journal receiver needed for media recovery of the queue. On queue managers on which circular logging is in place, this attribute has no value. This attribute is valid on UNIX, Linux, and Windows.	MONQ

Attribute	Meaning	MQSC parameter
Queue time	<p>The interval, in microseconds, between messages being put on the queue and then being destructively read. The maximum displayable value is 999 999 999; if the interval exceeds this value, 999 999 999 is displayed. The interval is measured from the time that the message is placed on the queue until it is destructively retrieved by an application and includes the following times:</p> <ul style="list-style-type: none"> • Time when the putting application has put the message, and the time when the message is available, for example, the message is committed. • Time the message is on the queue waiting to be processed. <p>The cell displays two values: a value based on recent activity over a short period of time, and a value based on activity over a longer period of time. These values depend on the configuration and behavior of your system and the levels of activity within it, and serve as an indicator that your system is performing normally. A significant variation in these values might indicate a problem with your system.  For z/OS queues that have the disposition Shared, the value shown is for measurements collected on this queue manager only.</p>	QTIME
Oldest message age	The age, in seconds, of the oldest message on the queue.	MSGAGE
Last put date	<p>The date on which the last message was put to the queue since the queue manager started. When no put date is available, perhaps because no message has been put to the queue since the queue manager was started, the value is shown as a blank.  For z/OS queues that have the disposition Shared, the value shown is for measurements collected on this queue manager only.</p>	LPUTDATE

Attribute	Meaning	MQSC parameter
Last put time	<p>The time at which the last message was put to the queue since the queue manager started. When no put time is available, perhaps because no message has been put to the queue since the queue manager was started, the value is shown as a blank.</p> <p>► z/OS For z/OS queues that have the disposition Shared, the value shown is for measurements collected on this queue manager only.</p>	LPUTTIME
Last get date	<p>The date on which the last message was retrieved from the queue since the queue manager started. A message being browsed does not count as a message being retrieved. When no get date is available, perhaps because no message has been retrieved from the queue since the queue manager was started, the value is shown as a blank.</p> <p>► z/OS For z/OS queues that have the disposition Shared, the value shown is for measurements collected on this queue manager only.</p>	LGETDATE
Last get time	<p>The time at which the last message was retrieved from the queue since the queue manager started. A message being browsed does not count as a message being retrieved. When no get time is available, perhaps because no message has been retrieved from the queue since the queue manager was started, the value is shown as a blank.</p> <p>► z/OS For z/OS queues that have the disposition Shared, the value shown is for measurements collected on this queue manager only.</p>	LGETTIME





Queue handle status objects

The following table lists the queue handle status attributes, which are shown in the second table in the Queue status dialog.









Attribute	Meaning	MQSC parameter
Queue name	The name of the queue.	

Attribute	Meaning	MQSC parameter
Application name	<p>A string containing the tag of the application connected to the queue manager. It is one of the following:</p> <ul style="list-style-type: none"> ▶ z/OS a z/OS batch job name • a TSO USERID, a CICS APPLID ▶ z/OS an IMS region name • a channel initiator job name ▶ IBM i an IBM i job name • a UNIX, Linux, and Windows process <p>The application name represents the name of the process or job that has connected to the queue manager. In the instance that this process or job is connected through a channel, the application name represents the remote process or job rather than the local channel process or job name.</p>	APPLTAG
▶ z/OS ▶ z/OS Process ID	The identifier of the process that opened the queue. This attribute is not valid on z/OS.	TID
▶ z/OS ▶ z/OS Thread ID	The identifier of the thread within the application process that has opened the queue. An asterisk indicates that this queue was opened with a shared connection. This attribute is not valid on z/OS.	
Application type	A string indicating the type of the application that is connected to the queue manager. Batch means an application using a batch connection; RRSBATCH means an RRS-coordinated application using a batch connection; CICS means a CICS transaction; IMS means an IMS transaction; CHINIT means a channel initiator; System means a queue manager; User means a user application.	APPLTYPE

Attribute	Meaning	MQSC parameter
Application description	A string containing a description of the application, where it is known, connected to the queue manager. If the application is not recognized by the queue manager the description returned consists of blanks. The application description is filterable (for example with a WHERE clause) to allow an administrator to display only certain connections.	APPLDESC
Browse access	This indicates whether the handle is providing browse access to the queue. If the handle is providing browse access, the value is Yes; if the handle is not providing browse access, the value is No.	BROWSE
Inquire access	This indicates whether the handle is providing inquire access to the queue. If the handle is providing inquire access, the value is Yes; if the handle is not providing inquire access, the value is No.	INQUIRE
Input access	This indicates whether the handle is providing input access to the queue. No means that the queue is not open for input; Shared means that the queue is open for shared input; Exclusive means that the queue is open for exclusive input.	INPUT
Output access	This indicates whether the handle is providing output access to the queue. If the handle is providing output access, the value is Yes; if the handle is not providing output access, the value is No.	OUTPUT
Set access	This indicates whether the handle is providing set access to the queue. If the handle is providing set access, the value is Yes; if the handle is not providing set access, the value is No.	SET
User ID	The user identifier that is associated with the handle.	USERID
Channel name	The name of the channel that owns the handle. If there is no channel associated with the handle, this value is empty. This value is shown only when the handle belongs to the channel initiator.	CHANNEL

Attribute	Meaning	MQSC parameter
Connection name	The connection name that is associated with the channel that owns the handle. If there is no channel associated with the handle, this value is empty. This value is shown only when the handle belongs to the channel initiator.	CONNNAME
Unit of work type	The type of unit of recovery as seen by the queue manager. It is one of the following:  CICS (z/OS only); XA;  RRS (z/OS only);  IMS (z/OS only); Queue manager.	URTYPE
Queue manager unit of work ID	The unit of recovery assigned by the queue manager. This is an 8 byte transaction identifier, displayed as 16 hexadecimal characters.  On z/OS, this is an 8 byte log RBA, displayed as 16 hexadecimal characters.	QMURID

Attribute	Meaning	MQSC parameter
Asynchronous state	<p>The state of the asynchronous consumer on this object handle. There are five possible values:</p> <p>Active: An MQCB call has set up a function to call-back to process messages asynchronously and the connection handle has been started so that asynchronous message consumption can proceed.</p> <p>Inactive: An MQCB call has set up a function to call-back to process messages asynchronously but the connection handle has not yet been started, or has been stopped or suspended, so that asynchronous message consumption cannot currently proceed.</p> <p>Suspended: The asynchronous consumption call-back has been suspended so that asynchronous message consumption cannot currently proceed on this object handle. This can be either because an MQCB call with Operation MQOP_SUSPEND has been issued against this object handle by the application, or because it has been suspended by the system. If it has been suspended by the system, as part of the process of suspending asynchronous message consumption the call-back function is called with the reason code that describes the problem resulting in suspension. This is reported in the Reason field in the MQCBC structure that is passed to the call-back function. For asynchronous message consumption to proceed, the application must issue an MQCB call with the Operation parameter set to MQOP_RESUME.</p> <p>Susptemp: The asynchronous consumption call-back has been temporarily suspended by the system so that asynchronous message consumption cannot currently proceed on this object handle. As part of the process of suspending asynchronous message consumption, the call-back function is called with the reason code that describes the problem resulting in suspension. This is reported in the Reason field in the MQCBC structure passed to the call-back function.</p>	ASTATE

Attribute	Meaning	MQSC parameter
External unit of work ID	The external unit of recovery identifier associated with the connection. It is the recovery identifier known in the external sync point coordinator. Its format is determined by the value of the Unit Of Work type attribute.	URID
  Address-space ID	A 4-character address-space identifier of the application that is identified by the Application name attribute. It distinguishes duplicate values of Application name. This value is displayed only when the queue manager that owns the queue is running on z/OS, and the Application type attribute does not have the value System.	ASID
  Program specification block name	The 8-character name of the program specification block (PSB) associated with the running IMS transaction (z/OS only). You can use the Program specification block name and Program specification table ID attributes to purge the transaction using IMS commands. A value is displayed only when the Application type attribute has the value IMS.	PSBNAME
  Program specification table ID	The 4-character IMS program specification table (PST) region identifier for the connected IMS region (z/OS only). A value is displayed only when the App type attribute has the value IMS.	PSTID
  CICS transaction ID	A 4-character CICS transaction identifier (z/OS only). A value is displayed only when the App type attribute has the value CICS.	TRANSID

Related concepts

[“IBM MQ queues” on page 29](#)

A queue is a container for messages. Business applications that are connected to the queue manager that hosts the queue can retrieve messages from the queue or can put messages on the queue.

Related tasks

[“Viewing the status of objects” on page 202](#)

You can display the current status of any object that can be in different states, in IBM MQ Explorer. For IBM MQ channels, you can also view the saved status.

Topic status attributes

The status attributes of topics.

For each attribute, there is a brief description of what information the attribute shows. The table also gives the equivalent MQSC parameter for the DISPLAY TPSTATUS command. For more information about MQSC commands, see [Script \(MQSC\) Commands](#) in the IBM MQ online product documentation.

Attribute	Meaning	MQSC parameter
Topic String	The Topic String identifies the topic node. It matches information from a publisher to a subscriber interested in that information.	TOPICSTR
Publish	Indicates whether publications are allowed or not.	PUB
Subscribe	Indicates whether subscriptions are allowed or not.	SUB
Durable subscriptions	Indicates whether durable subscriptions are allowed or not.	DURSUB
Default priority	Displays the default priority of messages published to the topic.	DEFPRTY
Default persistence	Displays the default persistence of messages published to the topic.	DEFPSIST
Model durable queue	This is the managed model queue for durable subscriptions.	MDURMDL
Model non-durable queue	This is the managed model queue for non-durable subscriptions.	MNDURMDL
Default put response type	The default response type for message puts. The default value is As parent. The 2 other options available are: Sync which means the response is put synchronously. Async which means the response is put asynchronously.	DEFPRESP
Admin topic name	Administrative topic objects are a required in order to be able to define attributes for certain portions of the topic tree, and to set up authority checking on specific topics.	N/A
Subscriber count	This is the number of subscribers for this topic string, including durable subscribers who are not currently connected.	SUBCOUNT
Publisher count	The number of applications currently publishing to the topic.	PUBCOUNT

Attribute	Meaning	MQSC parameter
Retained publication	Indicates if the publication is retained or not.	MQIACF_RETAINED_PUBLICATION
Non-persistent message delivery	The delivery method for non-persistent messages published to this topic.	NPMMSGDLV
Persistent message delivery	The delivery method for persistent messages published to this topic.	PMSGDLV
Publication scope	<p>The scope of publications can be controlled administratively using the PUBSCOPE topic attribute. The attribute can be set to one of the following 3 values:</p> <ul style="list-style-type: none"> • <code>As parent</code>. This is the default value. The publication scope is set to the same value as the parent queue manager. • <code>Queue manager</code>. The publication is only delivered to local subscribers. • <code>All</code>. The publication is delivered to local subscribers and remote subscribers by directly connected queue managers. 	PUBSCOPE
Subscription scope	<p>The scope of subscriptions can be controlled administratively using the SUBSCOPE topic attribute. The attribute can be set to one of the following 3 values:</p> <ul style="list-style-type: none"> • <code>As parent</code>. This is the default value. The subscription scope is set to the same value as the parent queue manager. • <code>Queue manager</code>. The subscription receives only local publications, and proxy subscriptions are not propagated to remote queue managers. • <code>All</code>. A proxy subscription is propagated to remote queue managers, and the subscriber receives local and remote publications. 	SUBSCOPE
Cluster name	This is the name of the cluster to which the topic belongs.	CLUSTER

Attribute	Meaning	MQSC parameter
Use dead-letter queue	<p>Specifies whether the dead-letter queue is used when publication messages cannot be delivered to their correct subscriber queue. There are 2 possible values:</p> <ul style="list-style-type: none"> • No means that publication messages that cannot be delivered to their correct subscriber queue are treated as a failure to put the message, and the application's MQPUT to a topic fails in accordance with the settings of Non-persistent message delivery and Persistent message delivery. • Yes means that if the queue manager Dead-letter queue attribute provides the name of a Dead Letter Queue, then it is used. Otherwise the behavior is as for No. 	USEDLQ

Related concepts

[“Topics” on page 30](#)

A topic identifies what a publication is about. A topic is a character string that describes the subject of the information that is published in a Publish/Subscribe message. As a subscriber, you can specify a topic or range of topics by using wildcards to receive the information that you require.

Related tasks

[“Viewing the status of objects” on page 202](#)

You can display the current status of any object that can be in different states, in IBM MQ Explorer. For IBM MQ channels, you can also view the saved status.

Related reference

[“Status attributes” on page 604](#)

In IBM MQ Explorer, you can view the current status of IBM MQ objects. For example, you can find out whether a channel is running, or you can find out when the last message was put on a certain queue. You can also view the saved status of a channel.

Subscription status attributes

The status attributes of subscriptions.

For each attribute, there is a brief description of what information the attribute shows. The table also gives the equivalent MQSC parameter for the DISPLAY QSTATUS command. For more information about MQSC commands, see [Script \(MQSC\) Commands](#) in the IBM MQ online product documentation.

Attribute	Meaning	MQSC parameter
Name	This is the application's unique subscription identifier.	SUB
Identifier	Identifier is assigned by the queue manager as a unique identifier for this subscription.	SUBID

Attribute	Meaning	MQSC parameter
User Id	The user identifier that is associated with the subscription.	SUBUSER
Durable	The subscription's Durable parameter can be either Yes or No. When Durable is set to Yes, then the subscriptions are not deleted when the creating application closes its subscription handle.	DURABLE
Type	<p>The Type of subscription indicates how the subscription was created. The subscription types are:</p> <p>API: Subscription created by using an MQSUB API request.</p> <p>ADMIN: Subscription created by using a DEF SUB MQSC or PCF command. ADMIN is also used to indicate that a subscription has been modified by using an administrative command.</p> <p>PROXY: Subscription created internally for routing publications through a queue manager network.</p>	SUBTYPE
Connection ID	The currently active CONNID that has opened this subscription. It is used to detect local publications.	ACTCONN
Resume date	The date of the most recent MQSUB which connected to this subscription.	RESMDATE
Resume time	The time of the most recent MQSUB which connected to this subscription.	RESMTIME
Date of last publication	The date on which a message was last sent to the destination specified by the subscription.	LMSGDATE
Time of last publication	The time on which a message was last sent to the destination specified by the subscription.	LMSGTIME

Attribute	Meaning	MQSC parameter
Message count	The number of messages put to the destination specified by this subscription since it was created, or since the queue manager was restarted, whichever is more recent. This number might not reflect the total number of messages that are, or have been, available to the consuming application. This is because it might also include publications that were partially processed but then undone by the queue manager due to a publication failure, or publications that were made within syncpoint that were rolled-back by the publishing application.	NUMMSGs
Multicast reliability indicator (%)	<p>Indicator of the reliability of the multicast messages. The values are expressed as a percentage. A value of 100 indicates that all messages are being delivered without problems. A value less than 100 indicates that some of the messages are experiencing network issues.</p> <p>To determine the nature of these issues, you can enable event message generation, using the COMMEV parameter of the COMMINFO objects, and examine the generated event messages. Two values are returned:</p> <ul style="list-style-type: none"> • The first value is based on recent activity over a short period of time. • The second value is based on activity over a longer period of time. If no measurement is available the values are shown as blanks. 	MCASTREL

Related tasks

[“Creating a new subscription” on page 137](#)

You can create a new subscription to subscribe to a topic for an IBM WebSphere MQ 7.0, or later, queue manager.

[“Viewing the status of objects” on page 202](#)

You can display the current status of any object that can be in different states, in IBM MQ Explorer. For IBM MQ channels, you can also view the saved status.

Related reference

[“Status attributes” on page 604](#)

In IBM MQ Explorer, you can view the current status of IBM MQ objects. For example, you can find out whether a channel is running, or you can find out when the last message was put on a certain queue. You can also view the saved status of a channel.

Topic status attributes for subscribers

For a given topic, the set of attributes that hold subscriber status information.

For each attribute, there is a brief description of what information the attribute shows. The table also gives the equivalent MQSC parameter for the DISPLAY TPSTATUS command. For more information about MQSC commands, see [Script \(MQSC\) Commands](#) in the IBM MQ online product documentation.

Attribute	Meaning	MQSC parameter
Topic String	The Topic String identifies the topic node. It matches information from a publisher to a subscriber interested in that information.	TOPICSTR
Subscription ID	Identifier is assigned by the queue manager as a unique identifier for the subscription.	SUBID
User	The user identifier that is associated with the subscription.	SUBUSER
Durable	Indicates whether durable subscriptions are allowed or not.	DURSUB
Type	The Type of subscription indicates how the subscription was created. The subscription types are: API: Subscription created via an MQSUB API request. ADMIN: Subscription created via a DEF SUB MQSC or PCF command. ADMIN is also used to indicate that a subscription has been modified via administrative command. PROXY: Subscription created internally for routing publications through a queue manager network.	SUBTYPE
Connection ID	The currently active CONNID that has opened this subscription. It is used to detect local publications.	ACTCONN
Resume date	The date of the most recent MQSUB which connected to this subscription.	RESMDATE
Resume time	The time of the most recent MQSUB which connected to this subscription.	RESMTIME

Attribute	Meaning	MQSC parameter
Message count	The number of messages put to the destination specified by this subscription since it was created, or since the queue manager was restarted, whichever is more recent. This number might not reflect the total number of messages that are, or have been, available to the consuming application. This is because it might also include publications that were partially processed but then undone by the queue manager due to a publication failure, or publications that were made within syncpoint that were rolled-back by the publishing application.	NUMMSGS
Multicast reliability indicator (%)	<p>Indicator of the reliability of the multicast messages. The values are expressed as a percentage. A value of 100 indicates that all messages are being delivered without problems. A value less than 100 indicates that some of the messages are experiencing network issues.</p> <p>To determine the nature of these issues, you can enable event message generation, using the COMMEV parameter of the COMMINFO objects, and examine the generated event messages. Two values are returned:</p> <ul style="list-style-type: none"> • The first value is based on recent activity over a short period of time. • The second value is based on activity over a longer period of time. If no measurement is available the values are shown as blanks. 	MCASTREL

Related concepts

[“Topics” on page 30](#)

A topic identifies what a publication is about. A topic is a character string that describes the subject of the information that is published in a Publish/Subscribe message. As a subscriber, you can specify a topic or range of topics by using wildcards to receive the information that you require.

Related tasks

[“Viewing the status of objects” on page 202](#)

You can display the current status of any object that can be in different states, in IBM MQ Explorer. For IBM MQ channels, you can also view the saved status.

Related reference

[“Status attributes” on page 604](#)

In IBM MQ Explorer, you can view the current status of IBM MQ objects. For example, you can find out whether a channel is running, or you can find out when the last message was put on a certain queue. You can also view the saved status of a channel.

[“Topic status attributes for publishers” on page 631](#)

For a given topic, the set of attributes that hold publication status information.

Topic status attributes for publishers

For a given topic, the set of attributes that hold publication status information.

For each attribute, there is a brief description of what information the attribute shows. The table also gives the equivalent MQSC parameter for the DISPLAY TPSTATUS command. For more information about MQSC commands, see [Script \(MQSC\) Commands](#) in the IBM MQ online product documentation.

Attribute	Meaning	MQSC parameter
Topic String	The Topic String identifies the topic node. It matches information from a publisher to a subscriber interested in that information.	TOPICSTR
Date of last publication	The date on which a message was last sent to the destination specified by the subscription.	LSMGDATE
Time of last publication	The time on which a message was last sent to the destination specified by the subscription.	LSMGTIME
Publish count	The number of applications currently publishing to the topic.	PUBCOUNT
Connection ID	The currently active CONNID that has opened a subscription. It is used to detect local publications.	ACTCONN

Attribute	Meaning	MQSC parameter
Multicast reliability indicator (%)	<p>Indicator of the reliability of the multicast messages. The values are expressed as a percentage. A value of 100 indicates that all messages are being delivered without problems. A value less than 100 indicates that some of the messages are experiencing network issues.</p> <p>To determine the nature of these issues, you can enable event message generation, using the COMMEV parameter of the COMMINFO objects, and examine the generated event messages. Two values are returned:</p> <ul style="list-style-type: none"> • The first value is based on recent activity over a short period of time. • The second value is based on activity over a longer period of time. If no measurement is available the values are shown as blanks. 	MCASTREL

Related concepts

[“Topics” on page 30](#)

A topic identifies what a publication is about. A topic is a character string that describes the subject of the information that is published in a Publish/Subscribe message. As a subscriber, you can specify a topic or range of topics by using wildcards to receive the information that you require.

Related tasks

[“Viewing the status of objects” on page 202](#)

You can display the current status of any object that can be in different states, in IBM MQ Explorer. For IBM MQ channels, you can also view the saved status.

Related reference

[“Status attributes” on page 604](#)

In IBM MQ Explorer, you can view the current status of IBM MQ objects. For example, you can find out whether a channel is running, or you can find out when the last message was put on a certain queue. You can also view the saved status of a channel.

[“Topic status attributes for subscribers” on page 629](#)

For a given topic, the set of attributes that hold subscriber status information.

Channel status attributes

Displays the attributes that show a channel's status. Two views are available: current status and saved status.




A channel's current status is updated continuously as messages are sent and received. A channel's saved status is updated only at the following times:

- For all channels:
 - When the channel enters or leaves Stopped or Retrying state
- For a sending channel:

- Before requesting confirmation that a batch of messages has been received
- When confirmation is received
- For a receiving channel:
 - Just before confirming that a batch of messages has been received
- For a server-connection channel:
 - No data is saved

Therefore, a channel that has never been current cannot have any saved status. The attributes that are displayed for a channel's saved status are a subset of the attributes that are displayed for a channel's status. These common attributes are marked with an asterisk (*) in the following table.


For each attribute, there is a brief description of what information the attribute shows. The table also gives the equivalent MQSC parameter for the DISPLAY CHSTATUS command. For more information about MQSC commands, see [Script \(MQSC\) commands](#).




Attribute	Meaning	MQSC parameter
Batch size	<p>The batch size that is being used for this session, valid only on the following platforms:</p> <ul style="list-style-type: none"> •  AIX •  HP-UX •  Linux • OS/400 •  Solaris •  Windows •  z/OS 	BATCHSZ
Batches	The number of batches completed since the channel started.	BATCHES
Buffers received	The number of transmission buffers received. This includes transmissions to receive control information only.	BUFSRCVD
Buffers sent	The number of transmission buffers sent. This includes transmission to send control information only.	BUFSENT
Bytes received	The number of bytes received since the channel started. This includes control information received by the Message Channel Agent.	BYTSRCVD
Bytes sent	The number of bytes sent since the channel started. This includes control information sent by the Message Channel Agent.	BYTSENT
*Channel monitoring	The current level of monitoring data collection for the channel.	MONCHL
*Channel name	The name of the channel definition.	CHANNEL

Attribute	Meaning	MQSC parameter
*Channel status	The status of the channel, which can be Starting, Binding, Initializing, Running, Stopping, Retrying, Paused, Stopped, or Requesting.	STATUS
Channel substate	The action that the channel is currently performing.	SUBSTATE
*Channel type	The type of the channel, which can be Sender, Server, Receiver, Requester, Cluster-sender, Cluster-receiver, Server-connection.	CHLTYPE
Compression rate	The compression rate achieved, displayed to the nearest percent. This displays a short-term indicator and a long-term indicator. These values are reset every time the channel is started and are displayed only when the channel is running.	COMPRATE
Compression time	The amount of time per message, in microseconds, spent during compression or decompression. This attribute displays a short-term indicator and a long-term indicator. These values are reset every time the channel is started and are displayed only when the channel is running.	COMPTIME
*Conn name	The connection name for which status information is displayed for the channel.	CONNNAME
Current conversations	The number of conversations that are currently being shared over a particular TCP/IP client channel instance (socket).	This is ephemeral information, and has no associated MQSC parameter.
*Current LUWID	The logical unit of work identifier that is associated with the current batch, for a sending or a receiving channel. For a sending channel, when the channel is in doubt it is the LUWID of the in-doubt batch. For a saved channel instance, this parameter has meaningful information only if the channel instance is in doubt. However, the parameter value is still returned when requested, even if the channel instance is not in doubt. It is updated with the LUWID of the next batch when this is known.	CURLUWID

Attribute	Meaning	MQSC parameter
*Current messages	For a sending channel, this is the number of messages that have been sent in the current batch. The value is incremented as each message is sent, and when the channel becomes in doubt it is the number of messages that are in doubt. For a saved channel instance, this parameter has meaningful information only if the channel instance is in doubt. However, the parameter value is still returned when requested, even if the channel instance is not in doubt. For a receiving channel, it is the number of messages that have been received in the current batch. It is incremented as each message is received. The value is reset to zero, for both sending and receiving channels, when the batch is committed.	CURMSGs
*Current sequence number	For a sending channel, this is the message sequence number of the last message sent. It is updated as each message is sent, and when the channel becomes in doubt it is the message sequence number of the last message in the in-doubt batch. For a saved channel instance, this parameter has meaningful information only if the channel instance is in doubt. However, the parameter value is still returned when requested, even if the channel instance is not in doubt. For a receiving channel, it is the message sequence number of the last message that was received. It is updated as each message is received.	CURSEQNO

Attribute	Meaning	MQSC parameter
Exit time	The amount of time, displayed in microseconds, that each message spent processing user exits. The cell displays two values: a value based on recent activity over a short period of time, and a value based on activity over a longer period of time. These values depend on the configuration and behavior of your system, as well as the levels of activity within it, and serve as an indicator that your system is performing normally. A significant variation in these values may indicate a problem with your system. They are reset every time the channel is started and are displayed only when the channel is running.	EXITTIME
Header compression	Whether the header data that is sent by the channel is compressed. Two values are shown: The default header data compression value negotiated for the channel, and the header data compression value that was used for the last message that was sent. If no message has been sent through the channel, the second value is blank.	COMPHDR
Heartbeat interval	The heartbeat interval that is being used for this session.	HBINT
*Indoubt status	Whether the channel is currently in doubt. This is only YES while the sending Message Channel Agent is waiting for an acknowledgment that a batch of messages that it has sent has been successfully received. It is NO at all other times, including the period during which messages are being sent, but before an acknowledgment has been requested. For a receiving channel, the value is always NO.	INDOUBT

Attribute	Meaning	MQSC parameter
Keep alive interval	<p>The length of the keep alive interval, 0 - 99999. This property is ignored if the channel uses a transport type other than TCP or SPX. The TCP Keep alive property must be set to Yes on the Channels page of the Queue manager properties.</p> <p> On z/OS queue managers, the Keep alive interval property specifies the keep alive interval for the individual channel.</p> <p>On queue managers on other platforms, the Keep alive interval property is used only if the channel connects to a z/OS queue manager; to use the functionality provided by the Keep alive interval property, set the Keep alive interval property to Auto to use a value based on the negotiated heartbeat interval value.</p>	KAINTE
*Last LUWID	The number of the last logical unit of work that was committed by the channel.	LSTLUWID
Last message date	The date when the last message was sent or MQI call was handled.	LSTMSGDA
Last message time	The time when the last message was sent or MQI call was handled. For a sender or server, this is the time the last message (the last part of it if it was split) was sent. For a requester or receiver, it is the time the last message was put to its target queue. For a server-connection channel, it is the time when the last MQI call completed.	LSTMSGTI
*Last sequence number	The number of the last message in the last batch that was committed by the channel.	LSTSEQNO
Local address	The local communications address for the channel. The value depends on the transport type of the channel. Currently, only TCP/IP is supported.	LOCLADDR
Long retries left	The number of long retry wait start attempts left. This applies only to sender or server channels.	LONGRTS

Attribute	Meaning	MQSC parameter
Max conversations	<p>Only available on Server-connection channels and Client-connection channels. Specifies the maximum number of conversations that can be shared over a particular TCP/IP client channel instance (socket). The possible values are:</p> <p>0: Specifies no sharing of conversations over a TCP/IP socket. The channel instance runs in a mode before that of IBM WebSphere MQ 7.0, with regard to:</p> <ul style="list-style-type: none"> • Administrator stop-quiesce • Heartbeating • Read ahead <p>1: Specifies no sharing of conversations over a TCP/IP socket. Client heartbeating and read ahead are available, whether in an MQGET call or not, and channel quiescing is more controllable.</p> <p>2 - 999999999: The number of shared conversations. The default value is 10.</p> <p>If the client-connection Max conversations value does not match the server-connection Max conversations value, then the lowest value is used.</p>	<p>SHARECNV</p> <p>(In MQSC this parameter is known as "Sharing conversations")</p>
MCA job name	<p>The name of the job currently serving the channel. The format depends on the platform:</p> <ul style="list-style-type: none"> •   On OS/400, UNIX systems, and Windows, this is the concatenation of the process identifier and the thread identifier of the MCA program displayed in hexadecimal. •  This information is not available on z/OS. 	JOBNAME
MCA status	The status of the Message Channel Agent, which is Running or Not running.	MCASTAT

Attribute	Meaning	MQSC parameter
MCA user ID	The user ID used by the MCA. This can be the user ID that is set in the channel definition, the default user ID for MCA channels, a user ID specified by a security exit, or, if the channel is a server-connection channel, a user ID transferred from a client.	MCAUSER
Message compression	The technique used to compress the message data sent by the channel. Two values are shown: the default message data compression value negotiated for the channel, and the message data compression value used for the last message that was sent. If no message has been sent through the channel, the second value is blank.	COMPMSG
*Messages	The number of messages that have been sent or received (or, for server-connection channels, the number of MQI calls handled) since the channel was started.	MSGS
Messages available	The number of messages that are queued on the transmission queue and are available to the channel for MQGETs.	XQMSGSA

Attribute	Meaning	MQSC parameter
Network time	<p>The amount of time, displayed in microseconds, to send an end of batch request to the remote end of the channel and receive a response. This is the time between sending the last message in a batch and receiving the end of batch acknowledgment, minus the processing time of the end of batch request at the remote end. There are two elements that make up the network time:</p> <ul style="list-style-type: none"> • The time that the data flowed on the network. • Delays at the remote end until the end of batch request is processed. If the channel processing at the remote end is delayed, for example, processing is retrying a put, then the processing of the end of batch request will be delayed and leads to an increase in NETTIME. <p>The cell displays two values: a value based on recent activity over a short period of time, and a value based on activity over a longer period of time. These values depend on the configuration and behavior of your system, as well as the levels of activity within it, and serve as an indicator that your system is performing normally. A significant variation in these values may indicate a problem with your system. They are reset every time the channel is started and are displayed only when the channel is running. This parameter applies only to sender, server, and cluster-sender channels.</p>	NETTIME
NPM speed	The nonpersistent message handling technique that is being used for this session.	NPMSPEED
Queue manager name	The name of the queue manager on which the channel is defined. For client-connection channels, this is the name of the queue manager to which an application that is running in the MQI client environment can request connection.	QMNAME

Attribute	Meaning	MQSC parameter
Remote product	The remote partner product identifier. This is the product identifier of the IBM MQ code running at the remote end of the channel. This field is available in IBM MQ 9.0 and later.	RPRODUCT
Remote queue manager	The queue manager name, or queue sharing group name, of the remote system.	RQMNAME
Remote version	The version of the IBM MQ code running at the remote end of the channel. If the remote version is blank, the remote partner is at version 6 or earlier.	RVERSION
Security protocol	Security protocol currently in use on the channel. Set automatically, based on the value you set for the SSL CipherSpec property. The value can be NONE, TLSV1, TLSV12.	SECPROT
*Short peer name	The Distinguished Name of the peer queue manager or client at the other end of the channel. The maximum length is 256 characters, so longer Distinguished Names are truncated.	SSLPEER
Short retries left	The number of short retry wait start attempts left. This applies only to sender or server channels.	SHORTRTS
SSL cert issuer name	The full Distinguished Name of the issuer of the remote certificate. The issuer is the Certificate Authority that issued the certificate. The maximum length is 256 characters, so longer Distinguished Names are truncated.	SSLCERTI
SSL cert user ID	The local user ID associated with the remote certificate.	SSLCERTU
SSL Cipher Spec	<p>The name of the Cipher Spec for a TLS connection. Both ends of the IBM MQ SSL channel definition must have the same value in the Cipher Spec property. For more information, see the SSLCIPH property in DEFINE CHANNEL.</p> <p>The value for this parameter is also used to set the value of the Security Protocol property.</p>	SSLCIPH

Attribute	Meaning	MQSC parameter
SSL key reset date	The date on which the previous successful TLS secret key was reset. The count of TLS secret key resets is reset when the channel instance ends. For more information, see Resetting SSL and TLS secret keys in the IBM MQ online product documentation.	SSLKEYDA
SSL key reset time	The time at which the previous successful TLS secret key was reset. The count of TLS secret key resets is reset when the channel instance ends. For more information, see Resetting SSL and TLS secret keys in the IBM MQ online product documentation.	SSLKEYTI
SSL key resets	The number of successful TLS key resets. The count of TLS secret key resets is reset when the channel instance ends. For more information, see Resetting SSL and TLS secret keys .	SSLRKEYS
Start date	The date when this channel started (in the form yyyy-mm-dd).	CHSTADA
Start time	The time when this channel started (in the form hh.mm.ss).	CHSTATI
Stop requested	Whether a user stop request is outstanding. The value is Yes or No.	STOPREQ
*Transmission queue	The name of the transmission queue for which status information is displayed for the specified channel.	XMITQ
Xmit batch size	The size of the batches transmitted over the channel. Two values are displayed: a value based on recent activity over a short period of time, and a value based on activity over a longer period of time. These values depend on the configuration and behavior of your system, as well as the levels of activity within it, and serve as an indicator that your system is performing normally. A significant variation in these values might indicate a problem with your system. The values are reset every time the channel is restarted and are displayed only when the channel is running.	XBATCHSZ

Attribute	Meaning	MQSC parameter
Xmit queue time	The time, in microseconds, that messages remained on the transmission queue before being retrieved. The time is measured from when the message is put on the transmission queue until it is retrieved to be sent on the channel and, therefore, includes any interval caused by a delay in the putting application. The cell displays two values: a value based on recent activity over a short period of time, and a value based on activity over a longer period of time. These values depend on the configuration and behavior of your system, as well as the levels of activity within it, and serve as an indicator that your system is performing normally. A significant variation in these values might indicate a problem with your system. The values are reset every time the channel is started and are displayed only when the channel is running.	XQTIME

Related concepts

[“Channels” on page 34](#)

IBM MQ can use three different types of channels: a message channel, an MQI channel, and an AMQP channel.

Related tasks

[“Viewing the status of objects” on page 202](#)

You can display the current status of any object that can be in different states, in IBM MQ Explorer. For IBM MQ channels, you can also view the saved status.

Listener status attributes

The status attributes of listeners.

For each attribute, there is a brief description of what information the attribute shows. The table also gives the equivalent MQSC parameter for the DISPLAY LSSTATUS command. For more information about MQSC commands, see [Script \(MQSC\) Commands](#) in the IBM MQ online product documentation.

Attributes	Meaning	MQSC parameter
Listener name	The name of the listener.	
Description	A descriptive comment of the listener.	DESCR
Listener status	The current status of the listener, which can be Running, Starting, or Stopping.	STATUS
PID	The operating system process identifier associated with the listener.	PID

Attributes	Meaning	MQSC parameter
Channel count	The current number of connections to the listener.	CURCONNS
Start date	The date on which the listener was started.	STARTDA
Start time	The time at which the listener was started.	STARTTI

Related concepts

[“Listeners” on page 38](#)

A listener is an IBM MQ process that listens for connections to the queue manager.

Related tasks

[“Viewing the status of objects” on page 202](#)

You can display the current status of any object that can be in different states, in IBM MQ Explorer. For IBM MQ channels, you can also view the saved status.

Service status attributes

The status attributes of custom services.

For each attribute, there is a brief description of what information the attribute shows. The table also gives the equivalent MQSC parameter for the DISPLAY SVSTATUS command. For more information about MQSC commands, see [Script \(MQSC\) Commands](#) in the IBM MQ online product documentation.

Attributes	Meaning	MQSC parameter
Service name	The name of the service.	
Description	A descriptive comment of the service.	DESCR
Service status	The current status of the service, which can be Running, Starting, or Stopping.	STATUS
PID	The operating system process identifier associated with the service.	PID
Start date	The date on which the service was started.	STARTDA
Start time	The time at which the service was started.	STARTTI

Related concepts

[“Custom services” on page 45](#)

Custom services are services that you create to run commands automatically.

Related tasks

[“Viewing the status of objects” on page 202](#)

You can display the current status of any object that can be in different states, in IBM MQ Explorer. For IBM MQ channels, you can also view the saved status.

Coupling facility structure status attributes

The status attributes of coupling facility (CF) structures.

For each attribute, there is a brief description of what information the attribute shows. The table also gives the equivalent MQSC parameter for the DISPLAY CFSTATUS command. For more information about MQSC commands, see [Script \(MQSC\) Commands](#) in the IBM MQ online product documentation.

Summary

This table lists the attributes in the Summary Status dialog, which displays the summary status information for the CF application structure.

Attribute	Meaning	MQSC parameter
Coupling facility name	The name of the CF structure.	
Type	The type of status information that is being displayed. Summary means that it is summary status information for the CF application structure; Connect means that it is connection status information for each CF application structure for each active queue manager; Backup means that it is backup status information for each CF application structure.	TYPE

Attribute	Meaning	MQSC parameter
Status	<p>The status of the CF application structure. If the value of Status type is Summary:</p> <ul style="list-style-type: none"> • Active means that the structure is active. • Failed means that the structure has failed. • Not Found means that the structure is not allocated in the coupling facility but has been defined to Db2. • Backup means that the structure is in the process of being backed up. • Recover means that the structure is in the process of being recovered. <p>If the value of Status type is Connect:</p> <ul style="list-style-type: none"> • Active means that the structure is active. • Failed means that the structure has failed. • None means that the structure has never been connected to this queue manager. <p>If the value of Status type is Backup:</p> <ul style="list-style-type: none"> • Active means that the structure is active. • Failed means that the structure has failed. • None means that the structure has never been backed up. • Backup means that the structure is in the process of being backed up. • In recover means that the structure is in the process of being recovered. 	STATUS
Max size	The size, in kilobytes, of the CF application structure.	SIZEMAX
Size used	The percentage of the CF application structure that is in use.	SIZEUSED
Max entries	The number of CF list entries defined for this CF application structure.	ENTSMAX

Attribute	Meaning	MQSC parameter
Entries used	The number of CF list entries defined for this CF application structure that are in use.	ENTSUSED
Fail date	The date on which this CF application structure failed. If Status type is Connect, this is the date on which the queue manager lost connectivity to this application structure. For the other values of Status type, this is the date on which this CF application structure failed. This value is shown only when the value of Status is Failed or In recover.	FAILDATE
Fail time	The time that this CF structure failed. If Status type is Connect, this is the time that the queue manager lost connectivity to this application structure. For the other values of Status type, this is the time that this CF application structure failed. This value is shown only when the value of Status is Failed or In recover.	FAILTIME
Offload use	This indicates whether offloaded large message data potentially exists in shared message data sets, Db2 or both. If Offload use is None, no offloaded large messages are present. In the case that Offload use is SMDS, offloaded large messages can exist in shared message data sets. Where Offload use is DB2, offloaded can exist in Db2. Finally if Offload use is Both, offloaded large messages can exist both in shared message data sets and in Db2.	

Connect

This table lists the attributes in the Connect Status dialog, which displays the connection status information for each CF application structure for each active queue manager.

Attribute	Meaning	MQSC parameter
Coupling facility name	The name of the CF structure.	
Queue manager name	The queue manager name.	QMNAME

Attribute	Meaning	MQSC parameter
System name	The name of the z/OS image of the queue manager that last connected to the CF application structure. These can be different across queue managers depending on your configuration setup.	SYSNAME
Status	<p>The status of the CF application structure. If the value of Status type is Summary:</p> <ul style="list-style-type: none"> • Active means that the structure is active. • Failed means that the structure has failed. • Not Found means that the structure is not allocated in the coupling facility but has been defined to Db2. • Backup means that the structure is in the process of being backed up. • Recover means that the structure is in the process of being recovered. <p>If the value of Status type is Connect:</p> <ul style="list-style-type: none"> • Active means that the structure is active. • Failed means that the structure has failed. • None means that the structure has never been connected to this queue manager. <p>If the value of Status type is Backup:</p> <ul style="list-style-type: none"> • Active means that the structure is active. • Failed means that the structure has failed. • None means that the structure has never been backed up. • Backup means that the structure is in the process of being backed up. • In recover means that the structure is in the process of being recovered. 	STATUS

Attribute	Meaning	MQSC parameter
Fail date	The date on which this CF application structure failed. If Status type is Connect, this is the date on which the queue manager lost connectivity to this application structure. For the other values of Status type, this is the date on which this CF application structure failed. This value is shown only when the value of Status is Failed or In recover.	FAILDATE
Fail time	The time that this CF structure failed. If Status type is Connect, this is the time that the queue manager lost connectivity to this application structure. For the other values of Status type, this is the time that this CF application structure failed. This value is shown only when the value of Status is Failed or In recover.	FAILTIME

Backup

This table lists the attributes in the Backup Status dialog, which displays the backup status information for each CF application structure.

Attribute	Meaning	MQSC parameter
Coupling facility name	The name of the CF structure.	
Queue manager name	The queue manager name.	QMNAME

Attribute	Meaning	MQSC parameter
Status	<p>The status of the CF application structure. If the value of Status type is Summary:</p> <ul style="list-style-type: none"> • Active means that the structure is active. • Failed means that the structure has failed. • Not Found means that the structure is not allocated in the coupling facility but has been defined to Db2. • Backup means that the structure is in the process of being backed up. • Recover means that the structure is in the process of being recovered. <p>If the value of Status type is Connect:</p> <ul style="list-style-type: none"> • Active means that the structure is active. • Failed means that the structure has failed. • None means that the structure has never been connected to this queue manager. <p>If the value of Status type is Backup:</p> <ul style="list-style-type: none"> • Active means that the structure is active. • Failed means that the structure has failed. • None means that the structure has never been backed up. • Backup means that the structure is in the process of being backed up. • In recover means that the structure is in the process of being recovered. 	STATUS
Backup date	The date on which the last successful backup was taken for this CF application structure.	BKUPDATE
Backup time	The end time of the last successful backup taken for this CF application structure.	BKUPTIME

Attribute	Meaning	MQSC parameter
Backup size	The size, in megabytes, of the last successful backup taken for this CF application structure.	BKSIZE
Start RBA	The backup data set start RBA for the start of the last successful backup taken for this CF application structure.	BKUPSRBA
End RBA	The backup data set end RBA for the end of the last successful backup taken for this CF structure.	BKUPERBA
Log queue manager name	A list of queue managers, the logs of which are required to perform a recover.	LOGS
Fail date	The date on which this CF application structure failed. If Status type is Connect, this is the date on which the queue manager lost connectivity to this application structure. For the other values of Status type, this is the date on which this CF application structure failed. This value is shown only when the value of Status is Failed or In recover.	FAILDATE
Fail time	The time that this CF structure failed. If Status type is Connect, this is the time that the queue manager lost connectivity to this application structure. For the other values of Status type, this is the time that this CF application structure failed. This value is shown only when the value of Status is Failed or In recover.	FAILTIME

SMDS

This table lists the attributes in the Backup Status dialog, which displays the backup status information for each CF application structure.

Attribute	Meaning	MQSC parameter
Coupling facility name	The name of the CF structure.	
Queue manager name	The queue manager name.	QMNAME
Access	The current availability state of the shared message data set. Valid availability states are Enabled, Suspended or Disabled	ACCESS

Attribute	Meaning	MQSC parameter
Fail date	The date on which this CF application structure failed. If Status type is Connect, this is the date on which the queue manager lost connectivity to this application structure. For the other values of Status type, this is the date on which this CF application structure failed. This value is shown only when the value of Status is Failed or In recover.	FAILDATE
Fail time	The time that this CF structure failed. If Status type is Connect, this is the time that the queue manager lost connectivity to this application structure. For the other values of Status type, this is the time that this CF application structure failed. This value is shown only when the value of Status is Failed or In recover.	FAILTIME
Recovery date	The recovery start date. If recovery is currently enabled for the data set, this indicates the date when it was activated, in the form yyyy-mm-dd.	RCVDATE
Recovery time	The recovery start time. If recovery is currently enabled for the data set, this indicated the time when it was activated, in the form hh.mm.ss,	RCVTIME

Attribute	Meaning	MQSC parameter
Status	<p>The status of the CF application structure. If the value of Status type is Summary:</p> <ul style="list-style-type: none"> • Active means that the structure is active. • Failed means that the structure has failed. • Not Found means that the structure is not allocated in the coupling facility but has been defined to Db2. • Backup means that the structure is in the process of being backed up. • Recover means that the structure is in the process of being recovered. <p>If the value of Status type is Connect:</p> <ul style="list-style-type: none"> • Active means that the structure is active. • Failed means that the structure has failed. • None means that the structure has never been connected to this queue manager. <p>If the value of Status type is Backup:</p> <ul style="list-style-type: none"> • Active means that the structure is active. • Failed means that the structure has failed. • None means that the structure has never been backed up. • Backup means that the structure is in the process of being backed up. • In recover means that the structure is in the process of being recovered. 	STATUS

Related concepts

[“Coupling facility structures” on page 47](#)

The coupling facility objects in IBM MQ Explorer represent coupling facility structures on a physical coupling facility. Coupling facility structures store the messages that are on shared queues. Each coupling facility structure used by IBM MQ is dedicated to a specific queue sharing group, but a coupling facility can hold structures for more than one queue sharing group.

Related tasks

[“Viewing the status of objects” on page 202](#)

You can display the current status of any object that can be in different states, in IBM MQ Explorer. For IBM MQ channels, you can also view the saved status.

Display SMDS status attributes

Information about the interaction between the shared message data sets (SMDS) for the named structure and the queue manager. The displayed properties are read only properties.

Display SMDS

This table lists the read only properties that are shown on the **Display SMDS** page of the coupling facility structures dialog.

Property	Meaning	MQSC parameter
Queue manager name	Read only: Name of the queue manager associated with the shared message data set.	SMDS
Coupling facility name	Read only: Name of coupling facility associated with the shared message data set.	CFSTRUCT
Number of buffers	Read only: This value displays the current setting on buffers allocated for accessing shared message data sets.	DSBUFS
Expand data set	Read only: This value provides information on the current setting for the expansion of the data set.	DSEXPAND

Display SMDS connections

This table lists the read only properties that are shown on the **Display SMDS connections** page of the coupling facility structures dialog.

Property	Meaning	MQSC Parameter
Queue manager name	Read only: Name of queue manager associated with the shared message data set.	SMDSCONN
Coupling facility name	Read only: Name of coupling facility associated with the shared message data set.	CFSTRUCT
Availability	Read only: This value displays the availability of the data set connection as seen by the queue manager.	AVAIL
Expansion status	Read only: This value displays the data sets automatic expansion status.	EXPANDST
Open mode	Read only: This value displays the mode in which the data set is currently open by the queue manager.	OPENMODE

Property	Meaning	MQSC Parameter
Status	Read only: This value displays the connection status as seen by the queue manager.	STATUS

Related tasks

[“Configuring queue managers and objects” on page 51](#)

You can configure many of the attributes of queue managers and their objects from IBM MQ Explorer using the properties dialogs.

Related reference

[“Strings in property dialogs” on page 655](#)

To include certain characters in a string, you must punctuate the string in a particular way.

Byte array dialog

The **Byte array** dialog is used to define or edit a byte array property of an IBM MQ object.

The **Byte array** dialog is launched from several sources, for example: from the **Subscription properties** dialog. The length of the byte array varies depending on which property you are defining, for example: A subscription's Correl ID is a maximum of 24 bytes long, but a subscription's Accounting token is a maximum of 32 bytes long.

When using the **Byte array** dialog, you have the choice of defining the array by entering text or bytes.

Attribute	Meaning
Text	The byte array appears as text in this field. If you want to edit or define the text, then edit this field.
Bytes	The byte array appears as bytes in this field. If you want to edit or define the bytes, then edit this field.

Related concepts

[“Objects in IBM MQ Explorer” on page 27](#)

In IBM MQ Explorer, all of the queue managers and their IBM MQ objects are organized in folders in the Navigator view.

Related tasks

[“Configuring queue managers and objects” on page 51](#)

You can configure many of the attributes of queue managers and their objects from IBM MQ Explorer using the properties dialogs.

Related reference

[“IBM MQ subscription properties” on page 483](#)

You can set attributes for all types of subscriptions. Some of the attributes do not apply to all types of subscriptions, some attributes are specific to z/OS subscriptions.

Strings in property dialogs

To include certain characters in a string, you must punctuate the string in a particular way.

Some of the properties you can set in the property pages are strings, in particular, the **Description**, **Cluster name**, and **Cluster namelist** properties.

To include certain characters (space, comma (,), single quotation marks (' '), and double quotation marks (" ")), you must punctuate the string in special ways.

Punctuate **Description** strings as follows:

- To enter a description without any commas or quotation marks, just type the string. For example, My queue

- To include a comma in a description, enclose the whole string in either single or double quotation marks. For example, "Beware, this is John's queue"
- To include quotation marks, either enclose the string in the other type of quotation marks, or repeat the quotation marks twice. For example, "Beware, this is John's ""special"" queue"

Punctuate **Cluster name** and **Namelist** properties as follows:

- Do not use a space or a comma in a cluster name. Use an underscore (_) instead. For example, `cluster_1`
- Type lists of cluster names separated by either spaces or commas. For example, `cluster_1 cluster_2 cluster_3,cluster_4` When the namelist is viewed, the delimiters are all commas and no spaces. So the example looks like this: `cluster_1,cluster_2,cluster_3,cluster_4`

Extending IBM MQ Explorer

You can extend the IBM MQ Explorer by writing one, or more, Eclipse plug-ins.

The IBM MQ Explorer is Eclipse based, and as such acquires all its functionality, and perspective information, through various plug-ins supplied with IBM MQ. To extend the IBM MQ Explorer you must to write one, or more, Eclipse plug-ins. Through writing a plug-in you can extend the function of the IBM MQ Explorer in the following ways:

- Add further menu options to existing pop-up menus and associate actions with them.
- Add tree nodes to the navigation view, and associated content pages.

When writing a plugin, you need to supply the following resources:

A **plugin.xml** file

Specify extension points in a `plugin.xml` file. Use the extension points to extend the functionality of the IBM MQ Explorer. There are many types of extension point available within the IBM MQ Explorer and Eclipse. Each type of extension point is used to extend the IBM MQ Explorer in a different way. Most extension points are associated with a Java archive (JAR) file. For more information on the extension points available, see [“Utilizing extension points”](#) on page 658.

A set of Java archive (JAR) files

Write classes that implement the functionality specified by the extension points that you declared in the `plugin.xml` file. Package the classes in JAR files, so that every JAR file is associated with at least one extension point.

IBM MQ supplies sample Eclipse plug-ins called `simple`, and `menu`. The `simple` plug-in uses all the extension points provided in the IBM MQ Explorer to extend the Explorer in a number of basic ways. The `simple` plug-in can be used as a basis for writing your own Eclipse plug-ins. For instructions of how to import the `simple` plugin, see [“Writing an Eclipse plug-in for IBM MQ Explorer”](#) on page 657.

Information on how to write Eclipse plug-ins is available in the *Platform Plug-in Developers Guide*, found in the online Eclipse help. See <https://help.eclipse.org/latest/index.jsp?nav=%2F2> for more information.

Related concepts

[“Writing an Eclipse plug-in for IBM MQ Explorer”](#) on page 657

How to write an Eclipse plug-in for IBM MQ Explorer by using the extension points that are available to extend the IBM MQ Explorer functionality.

[“Applying plug-ins to IBM MQ Explorer”](#) on page 662

You can either run a plug-in with IBM MQ Explorer from the Eclipse workbench, or apply updates from a plug-in to IBM MQ Explorer permanently.

Importing the sample Eclipse plug-ins

Instructions for importing the sample Eclipse plug-ins.

To import the sample Eclipse plug-ins, complete the following steps:

1. Install the IBM MQ Explorer into an Eclipse environment, as described in [“Installing into Eclipse environments”](#) on page 24
2. Open the **Plug-in Development** perspective.
3. Click **File > Import** to open the Import wizard.

In the Import wizard, complete the following steps:

- a. Click **Plug-in Development > Plug-ins and Fragments**.
 - b. Select the **Projects with source folders** check box and click **Next**.
 - c. Select one or more from:
 - com.ibm.mq.explorer.sample.simple
 - com.ibm.mq.explorer.sample.menus
 - com.ibm.mq.explorer.jmsadmin.sample.menus
 - com.ibm.mq.explorer.tests.sample
 - d. Click the **Add** button, then click **Finish**.
4. If you selected com.ibm.mq.explorer.tests.sample in the previous step you need to returned to the Import wizard and complete the following steps:
 - a. Click **Plug-in Development > Plug-ins and Fragments**.
 - b. Select the **Binary projects** check box and click **Next**.
 - c. Select the com.ibm.mq.runtime plugin.
 - d. Click the **Add** button, then click **Finish**.

You have now imported the sample Eclipse plug-ins.

Writing an Eclipse plug-in for IBM MQ Explorer

How to write an Eclipse plug-in for IBM MQ Explorer by using the extension points that are available to extend the IBM MQ Explorer functionality.

Note: It is assumed that you have the prerequisite knowledge detailed in [“Extending IBM MQ Explorer”](#) on page 656.

To write an Eclipse plug-in for the IBM MQ Explorer, you must use the extension points available to extend the functionality of the IBM MQ Explorer. The most common extension points are described, and accompanied by a number of code extracts from the simple plug-in to provide basic implementation examples. You must import the simple plug-in if you want access to the code that it contains. For instructions on how to import the simple plugin, see [“Importing the sample Eclipse plug-ins”](#) on page 656.

The environment in which the IBM MQ Explorer is extended is an event driven interface. For example, when a Register extension point is extended with an instance of a user-written class that extends the IExplorerNotify interface, the user-written class will be called back when an event occurs. For example, when a queue manager is created. Many of these notifications include a MQExtObject as one of their arguments. An MQExtObject relates to the IBM MQ object that caused the event. A user-written class can call any of the MQExtObject public methods to find out about the object.

The IExplorerNotify interface, the associated MQExtObject, and other external definitions are documented in IBM MQ Explorer Javadoc documentation. For information on how to access the IBM MQ Explorer Javadoc documentation, see [“API Reference”](#) on page 662.

Related concepts

[“Utilizing extension points”](#) on page 658

Instructions on how to use the extension points available in Eclipse plug-ins for the IBM MQ Explorer.

Utilizing extension points

Instructions on how to use the extension points available in Eclipse plug-ins for the IBM MQ Explorer.

For further information on using extension points see <https://help.eclipse.org/latest/index.jsp?nav=%2F2> then select **Programmer's Guide**.

For information on how to include an extension point, see *Plugging into the workbench->Basic workbench extension points using actions in the Programmers Guide*.

Through utilizing the available extension points, you can extend the function of the IBM MQ Explorer in the following ways:

- Register extension points.
- Add further menu options to existing menus and associate actions with them.
- Add tree nodes to the navigation view and associate content pages with them.
- Add property tabs to property dialogs and associate property pages with them.

Multiple extension points of the same type can be included in a single plugin. The extension points that you use will be dependent on the way in which you intend to extend the functionality of the IBM MQ Explorer. However, every plug-in for the IBM MQ Explorer must use the register extension point.

Related concepts

[“Register a plugin-in with IBM MQ Explorer” on page 658](#)

How to use the register extension point to register a plugin-in with IBM MQ Explorer and enable notify events.

[“Add tree node” on page 659](#)

A tree node extension point is used to add a tree node to the navigation view and associate it with a content page.

[“Add content page” on page 660](#)

A content page extension point is used to add a content page to the content view. A content page can be associated with a tree node.

[“Add a pop-up menu item” on page 660](#)

You can use a pop-up menu extension point to add pop-up menu items to the IBM MQ Explorer.

[“Adding a property tab to an Eclipse property dialog” on page 661](#)

A property tab extension point is used to add a property tab to a property dialog and an associated property page.

Register a plugin-in with IBM MQ Explorer

How to use the register extension point to register a plugin-in with IBM MQ Explorer and enable notify events.

The register extension point is used for the following:

- To allow your plug-in to register itself with the IBM MQ Explorer. Every plug-in for the IBM MQ Explorer must include this extension point in `plugin.xml`. Without it, any function your plug-in adds to the IBM MQ Explorer will not be activated.
- To enable notify events.

The following code extract is taken from the file, `plugin.xml`, from the simple plug-in and shows a basic implementation of the register extension point:

```
<extension
    id="com.ibm.mq.explorer.sample.simple"
    name="Simple Sample"
    point="com.ibm.mq.explorer.ui.registerplugin">
  <pluginDetails
    pluginId="com.ibm.mq.explorer.sample.simple"
    name="Simple"
```

```

        class="com.ibm.mq.explorer.sample.simple.SimpleNotify"
        enabledByDefault="true"
        description="a very simple sample plug-in to Explorer"
        vendor="IBM">
    </pluginDetails>
</extension>

```

Related concepts

[“Enabling and disabling a plugin” on page 659](#)

How to enable and disable plugin-ins that contain the register extension point.

[“Notify events” on page 659](#)

Within the IBM MQ Explorer, when an IBM MQ object is created, or manipulated, a Java object relating to the IBM MQ object can be generated.

Enabling and disabling a plugin

How to enable and disable plugin-ins that contain the register extension point.

All plug-ins that contain the register extension point can be enabled, or disabled, within the IBM MQ Explorer by doing the following:

1. From the IBM MQ Explorer toolbar click, **Window -> Preferences**.
2. Expand **IBM MQ Explorer**.
3. Click **Enable plug-ins**.

All registered plug-ins are displayed.

4. Select all plug-ins that should be enabled.
5. Click **OK**.

Notify events

Within the IBM MQ Explorer, when an IBM MQ object is created, or manipulated, a Java object relating to the IBM MQ object can be generated.

These Java objects can be used to find the name, type, and other externalized attributes of an IBM MQ object.

For Java objects to be generated, the register extension point must specify a class. In the `plugin.xml` file from the simple plugin, the class specified is as follows:

```
class="com.ibm.mq.explorer.sample.simple.SimpleNotify"
```

This class contains a number of object specific methods. When an IBM MQ object is created, or manipulated, the appropriate method from the notify class is called. This class can be used as a basis for writing your own class. For the methods that this class must contain refer to the IBM MQ Explorer Javadoc documentation. For information on how to access the IBM MQ Explorer Javadoc documentation, see [“API Reference” on page 662](#).

Add tree node

A tree node extension point is used to add a tree node to the navigation view and associate it with a content page.

The following code extract is taken from the file, `plugin.xml`, from the simple plug-in and shows a basic implementation of the tree node extension point:

```

<extension
    id="com.ibm.mq.explorer.samples.simpleTreeNode"
    name="Simple TreeNode"
    point="com.ibm.mq.explorer.ui.addtreenode">
    <treeNode
        pluginId="com.ibm.mq.explorer.sample.simple"
        name="com.ibm.mq.explorer.sample.simple"
        class="com.ibm.mq.explorer.sample.simple.SimpleTreeNodeFactory"
        treeNodeId="com.ibm.mq.explorer.sample.simple"
        sequence="888">

```

```

        </treeNode>
    </extension>

```

As well as declaring the tree node extension point in plugin.xml, the following classes are needed:

- A class that contains a method that checks the ID of any incoming tree node to determine whether to add sub nodes to it. This class must implement com.ibm.mq.explorer.ui.extensions.ITreeNodeFactory, and IExecutableExtension. For the methods that this class must contain, refer to the IBM MQ Explorer Javadoc documentation. For information on how to access the IBM MQ Explorer Javadoc documentation, see [“API Reference” on page 662](#).

A working example of this class is available in the simple plugin, called SimpleTreeNodeFactory.java

- A class that contains methods that return information about any new tree nodes, such as the name, ID, and the associated content page class. This class must extend com.ibm.mq.ui.extensions.TreeNode. For the methods that this class must contain refer to the IBM MQ Explorer Javadoc.

A working example of this class is available in the simple plugin, called SimpleTreeNode.java.

Add content page

A content page extension point is used to add a content page to the content view. A content page can be associated with a tree node.

The following code extract is taken from the file, plugin.xml, from the simple plug-in and shows a basic implementation of the content page extension point:

```

<extension
    id="com.ibm.mq.explorer.sample.simpleContentPage"
    name="Simple ContentPage"
    point="com.ibm.mq.explorer.ui.addcontentpage">
    <contentPage
        pluginId="com.ibm.mq.explorer.sample.simple"
        name="com.ibm.mq.explorer.sample.simple"
        class="com.ibm.mq.explorer.sample.simple.SimpleContentPageFactory"
        contentPageId="com.ibm.mq.explorer.sample.simple">
    </contentPage>
</extension>

```

As well as declaring the content page extension point in plugin.xml, the following classes are needed:

- A class that contains methods that perform a number of functions such as return the content page id, create the content page, and set the object to draw the page. This class must extend com.ibm.mq.ui.extensions.ContentPage. The class com.ibm.mq.explorer.ui.extensions.ContentTitleBar can be used to create a title for the content page consistent with the other content pages in the IBM MQ Explorer. For the methods that this class must contain, refer to the IBM MQ Explorer Javadoc documentation. For information on how to access the IBM MQ Explorer Javadoc documentation, see [“API Reference” on page 662](#).

A working example of this class is available in the simple plugin, called SimpleContentPage.java.

- A class that contains a method that returns an instance of the class extending ContentPage. This class must implement com.ibm.mq.explorer.ui.extensions.IContentPageFactory, and IExecutableExtension. For the methods that this class must contain refer to the IBM MQ Explorer Javadoc documentation.

A working example of this class is available in the simple plugin, called SimpleContentPageFactory.java

Add a pop-up menu item

You can use a pop-up menu extension point to add pop-up menu items to the IBM MQ Explorer.

The following code extract is taken from the plugin.xml file, which you can find in the simple plugin, and shows a basic implementation of the pop-up menu extension point:

```

<extension
    id="com.ibm.mq.explorer.sample.simple.object1"
    name="Object1"
    point="org.eclipse.ui.popupMenus">
    <objectContribution

```

```

        objectClass="com.ibm.mq.explorer.ui.extensions.MQExtObject"
        id="com.ibm.mq.explorer.sample.simple.obj1">
    <visibility>
        <and>
            <pluginState
                value="activated"
                id="com.ibm.mq.explorer.ui">
            </pluginState>
            <objectClass
                name="com.ibm.mq.explorer.ui.extensions.MQExtObject">
            </objectClass>
            <objectState
                name="PluginEnabled"
                value="com.ibm.mq.explorer.sample.simple">
            </objectState>
        </and>
    </visibility>
    <action
        label="Simple: Sample action on any MQExtObject"
        class="com.ibm.mq.explorer.sample.simple.MenuActions"
        menubarPath="additions"
        id="com.ibm.mq.explorer.sample.simple.obj.action1">
    </action>
</objectContribution>
</extension>

```

You can add menu items by using the Eclipse Platform extension point `org.eclipse.ui.popupMenus`. The `<visibility>` attribute in the preceding extract contains the elements that control the conditions under which the pop-up menu item is displayed. These conditions include tests on the plug-in state, the type of object, and the state of the object. For example, a content menu item can be displayed for local queues only, or for remote queue managers only.

Adding a property tab to an Eclipse property dialog

A property tab extension point is used to add a property tab to a property dialog and an associated property page.

The following code extract is taken from the file `plugin.xml`, from the simple plugin, and shows a basic implementation of the property tab extension point:

```

<extension
    id="com.ibm.mq.explorer.samples.simplePropertyTab"
    name="Simple Property Tab"
    point="com.ibm.mq.explorer.ui.addpropertytab">
    <propertyTab
        class="com.ibm.mq.explorer.sample.simple.SimplePropertyTabFactory"
        objectId="com.ibm.mq.explorer.queuemanager"
        pluginId="com.ibm.mq.explorer.sample.simple"
        name="com.ibm.mq.explorer.sample.simple"
        propertyTabId="com.ibm.mq.explorer.sample.simple.propertyTab"
        propertyTabName="Simple Sample Property Tab"/>
    </propertyTab>
</extension>

```

As well as declaring the property tab extension point in `plugin.xml`, the following classes are needed:

- A class that contains a method that creates and returns a property page to be displayed when a user clicks the property tab. This class must implement `com.ibm.mq.explorer.ui.extensions.IPropertyTabFactory`. For the methods that this class must contain refer to the IBM MQ Explorer Javadoc documentation. For information on how to access the IBM MQ Explorer Javadoc documentation, see [“API Reference” on page 662](#).

A working example of this class, called `SimplePropertyTabFactory.java`, is available in the simple plugin.

- A class used for creating the property page must extend `com.ibm.mq.ui.extensions.PropertyPage`. For the methods that this class must contain refer to the IBM MQ Explorer Javadoc documentation.

A working example of this class, called `SimplePropertyPage.java`, is available in the simple plugin.

Applying plug-ins to IBM MQ Explorer

You can either run a plug-in with IBM MQ Explorer from the Eclipse workbench, or apply updates from a plug-in to IBM MQ Explorer permanently.

To run plug-ins with IBM MQ Explorer from the Eclipse workbench, complete the following steps:

1. Select the plug-in from the Package Explorer.
2. Click **Run > Run As > Eclipse Application**.

A new Eclipse workbench opens.

3. In the new Eclipse workbench, open the IBM MQ Explorer perspective.
4. In the **Explorer preferences** section, select the "Enable plug-ins" page and enable the relevant sample plug-in or plug-ins.

To permanently apply updates to IBM MQ Explorer provided by a plugin, complete the following steps:

1. With a file browser, find the plug-in file that provides the functionality extensions to IBM MQ Explorer.
2. Copy the plug-in file, and paste it into `MQExplorer\eclipse\dropins` within your IBM MQ installation directory. For example, on Windows: `C:\Program Files\IBM\WebSphere MQ\` or `/opt/mqm` on Linux x86-64 platforms.
3. Restart IBM MQ Explorer.

The updates provided by the plug-in are applied to IBM MQ Explorer.

API Reference

Reference information for the IBM MQ Explorer API.

The API Reference information is available only in the installed IBM MQ Explorer.

To access this information, Launch IBM MQ Explorer, then visit this topic in the embedded Help documentation.

Notices

This information was developed for products and services offered in the United States. IBM may not offer the products, services, or features discussed in this information in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this information. The furnishing of this information does not give you any license to these patents. You can send license inquiries, in writing, to:

- IBM Director of Licensing
- IBM Corporation
- North Castle Drive
- Armonk, NY 10504-1785
- U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

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 information. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this information at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

- IBM United Kingdom Laboratories,
- Mail Point 151,
- Hursley Park,
- Winchester,
- Hampshire,
- England
- SO21 2JN.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Programming License Agreement, or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

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. 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 IBM's application programming interfaces.

Trademarks

The following are trademarks of International Business Machines Corporation in the United States, or other countries, or both:

- IBM
- AIX
- CICS
- Db2
- IMS
- MQ
- MQSeries®
- MVS/ESA
- VSE/ESA
- OS/390®
- OS/400
- FFST
- First Failure Support Technology
- WebSphere
- z/OS
- i5/OS

Windows is a registered 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 trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

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 Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Software Interoperability Coordinator, Department 49XA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this 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 WebSphere 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 (<http://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: