Unit 10. Implementing basic security in IBM MQ

Estimated time

01;00

Overview

In this unit, you learn how IBM MQ protects its objects by using access control lists (ACLs), and how the IBM MQ Object Authority Manager (OAM) uses these ACLs when a user attempts to access these objects. The unit also describes how to manage IBM MQ object authorizations and introduces Secure Socket Layer (SSL) and Transport Layer Security (TLS) support.

How you will check your progress

- Review questions
- Hands-on exercise

References

IBM Knowledge Center for IBM MQ V9

IBM Training

Unit objectives

- Describe the role of the object authority manager (OAM) to provide security to IBM MQ resources
- Protect IBM MQ resources by using the OAM
- Use some of the OAM control commands
- Describe the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) support that IBM MQ provides
- Implement basic channel authentication

Figure 10-1. Unit objectives

The first topic provides a review on security. MQ security concepts were introduced in course WM103/ZM103, *Technical Introduction to IBM MQ V9*, which is a prerequisite for this course.

IBM Training

Security mechanisms

- **Identification**
  Uniquely identify users of a system or application that is running in the system
- **Authentication**
  Prove that a user or application is genuinely that person or what that application claims to be
- **Access control**
  Protect resources in a system by limiting access to only authorized users and their applications
- **Confidentiality**
  Encrypt messages to protect data
- **Auditing**
  Track users and applications that access the system

Figure 10-2. Security mechanisms

A comprehensive security implementation includes the following requirements:

- The ability to uniquely identify users of a system or application
- The ability to prove that a user or application is authentic
- The ability to protect resources by limiting access to authorized users and applications
- The ability to protect confidential data
- The ability to track users and applications that access the system and the data

This unit shows how MQ supports the security implementations.

IBM Training

IBM MQ security implementations

- OAM installable service
- SSL for channel security
- Channel authentication rules for channel access control
- Connection authorization for queue manager access control
- Connection refusal events

Figure 10-3. IBM MQ security implementations

This figure lists the MQ security implementations for identification, authentication, access control, confidentiality, and auditing.

The MQ Object Authority Manager (OAM) service provides **authorization** for MQI calls, commands, and access to objects to protect the local MQ resources.

The SSL protocol provides industry-standard channel security, with protection against eavesdropping, tampering, and impersonation to control access to the network. An MQ channel can be configured to receive and authenticate an SSL certificate from an SSL client.

By using MQ channel authentication, you can provide more precise control over the access that is granted to connecting systems at a channel level.

MQ connection authorization uses the supplied user ID and password to check whether a user has authority to access resources.

MQ security provides descriptive connection refusal events, which are written to a channel event queue.

IBM Training

Planning for security

1. Identify users that need authority to administer IBM MQ
2. Identify applications that need authority to work with IBM MQ objects
3. User ID that is associated with MCA's authority to access various IBM MQ resources

Figure 10-4. Planning for security

You can use MQ for a wide variety of applications on a range of operating systems and hardware. The security requirements are likely to be different for each application. The first step to any security implementation is to identify the security requirements.

You must consider certain aspects of security when you implement MQ. If you ignore security aspects and do nothing on some operating systems, such as IBM i, UNIX, Linux, and Windows, you cannot use MQ.

First, identify users that need the authority to administer MQ. These users are the users that need to be able to enter commands and use MQ Explorer to access queue managers, queues, channels, and processes.

Second, identify applications that need to access to MQ queue managers, queues, processes, namelists, and topics by using MQI calls.

Third, identify the user IDs that are associated with the applications that need to administer channels and channel initiators, and open transmission queues.

For more information about security planning, see the IBM MQ product documentation.

IBM Training

IBM MQ access control overview

- Granular access control facilities
  - Provided by IBM MQ installable services
  - Which user? Which resource? What types of access?
  - Channel access control that is based on IP address, queue manager, SSL distinguished name, and asserted identity
- IBM MQ access control at user and group level
- Alternative user IDs can be specified when suitably authorized
- User needs access to the first named resource and not the alias queue or remote queue resolved resource

Figure 10-5. IBM MQ access control overview

MQ access control is the primary security component. Access control allows MQ to control which users and applications are granted specific levels access to specific MQ resources. Resources that might be controlled in this way are the queue manager, queues, and processes.

All queue managers on distributed operating systems provide access control facilities to control which users have access to which MQ resources. These queue managers use the OAM component of MQ to provide access control for MQ resources. By using the OAM, the MQ administrator can control access at the user level and at the group level.

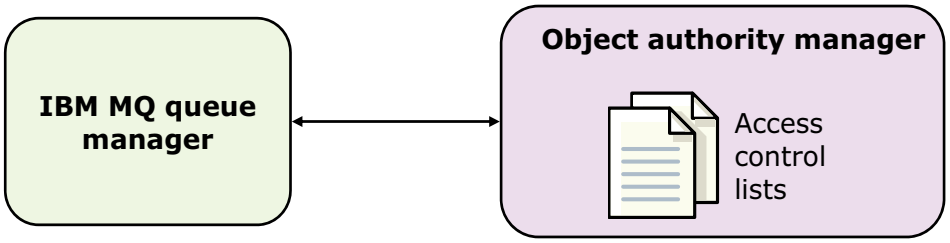
When an application attempts to connect to MQ, MQ captures the user ID that is associated with the application from the MQCONN call. This user ID is used for the access control checks.

It is possible for authorized users to use an alternative user ID instead of the logged on user ID. The name that is specified in the MQ API command is used when MQ checks to see whether a user is authorized to access a particular resource.

For an alias or remote queue definition, the application user ID needs access to the queue that is specified in the MQ API command and not the resolved name. For model queues, MQ might generate the name of the dynamic queue in some instances. In this case, the user ID creating a dynamic queue is automatically given full access rights to the queue.

IBM Training

OAM installable service



- IBM MQ authorization service component
- Common access control list (ACL) manager for distributed queue managers
- Authorizations can be granted or revoked at the principal or group level
- IBM MQ Explorer, control commands, and MQSC commands can be used to configure and manage ACLs

Figure 10-6. OAM installable service

The MQ OAM component is an MQ installable service.

The MQ OAM provides a full set of access control facilities for MQ including access control checking and commands to set, change, and inquire on MQ access control information.

Access to MQ entities is controlled through MQ user groups and the OAM.

The authorization service maintains an ACL for each MQ object to which it is controlling access. An ACL contains a list of all the group IDs and user IDs that can access the object.

Administrators can use MQ Explorer and commands to configure and manage ACL objects.

IBM Training

Principals and groups

- On UNIX and Linux
  - *Principal* is a user ID or an ID that is associated with an application program that is running on behalf of a user
  - *Group* is a system-defined collection of principals
  - ACLs are based on groups by default
  - Queue manager can be configured to support principals at creation with **`-oa user`** option or by editing **`qm.ini`**
- On Windows
  - *Principal* is a Windows user ID, or an ID that is associated with an application program that is running on behalf of a user
  - *Group* is a Windows group
  - ACLs are based on principals (user IDs) and groups
- Changes to a principal's group membership are not recognized until the queue manager is restarted or administrator runs **`REFRESH SECURITY`** MQSC command

Figure 10-7. Principals and groups

Users can belong to groups. You can grant access to a particular resource to groups rather than to individuals to reduce the amount of administration required. For example, you might define a group that consists of users who want to run a particular application. Other users can be given access to all the resources they require by adding their user ID to the appropriate group.

On UNIX and Linux, all ACLs are based on groups by default. You can change the queue manager to support principals, similar to Windows.

On Windows, ACLs are based on user IDs and groups.

Any changes that you make to a principal's group membership are not recognized until the queue manager is restarted, or you enter the **`REFRESH SECURITY`**. MQSC command.

IBM Training

Access control with the OAM

- Access control lists are specific to IBM MQ
  - Not integrated with system-level security
  - Changes to user's operating system authority are not recognized until queue manager restart or security refresh
  - One ACL for each queue manager, not shared between queue managers
- Use **`setmqaut`** control command, the **`SET AUTHREC`** MQSC command, and IBM MQ Explorer
  - Give users, and groups of users, access to IBM MQ objects
- Access control for IBM MQ objects
  - Queue manager
  - Queues
  - Processes
  - Namelists
  - Channels
  - Authentication information objects
  - Listeners

Figure 10-8. Access control with the OAM

Each queue manager contains a set of ACLs for each queue manager. ACLs cannot be (automatically) shared with multiple queue managers. The ACLs are not integrated with system-level security.

The OAM provides access control facilities only for MQ objects: the queue manager, all queues, processes, namelists, channels, authentication information objects, listeners, and services.

You can use the **`setmqaut`** control command, the **`SET AUTHREC`** MQSC command, or IBM MQ Explorer to give users, and groups of users, access to MQ objects.

Note

On an IBM MQ Appliance, you can use only the **`SET AUTHREC`** command.

IBM Training

OAM access control lists

- One authorization file for each object plus global permissions files
- Can reference LDAP repository by specifying principal and group names in LDAP form
- Windows OAM bypasses authorization files for certain classes of principal:
  - SYSTEM
  - Local `Administrators` group
  - Local `mqm` group
- Uses IBM MQ object authorizations
  - Context, such as `passall`, `passid`
  - MQI, such as `browse`, `put`, `get`, `set`
  - Administration, such as `dsp`, `chg`, `dlt`
  - Generic, such as `all`, `alladm`, `allmqi`, `none`

Figure 10-9. OAM access control lists

Each OAM authorization file contains a set of access control stanzas. One stanza exists per principal for which access is controlled, where a principal is either a user ID or a group. MQ can be configured to connect to an LDAP repository and reference principal and group names in LDAP form.

The principals (user IDs, groups, or both) that can access an object are listed in this file. This file includes a bit string (in hex) that represents the access rights that are associated with that entity.

Certain principals or groups are granted automatic access to MQ resources:

- Members of the “mqm” group or the “mqm” user
- On Windows:
  - Administrator user and local group
  - SYSTEM user ID
  - The user or principal group that creates a resource

Authorizations can be assigned to the following categories:

- Authorizations for MQI context such as passing all the context fields (`passall`) or the identity context (`passid`) from the request message to a message that the application is putting on the queue
- Authorizations for sending MQI calls to browse, put, or get messages, or to set queue attributes
- Authorizations for entering commands for administration tasks such as displaying (`dsp`), changing (`chg`), and deleting (`dlt`) objects
- Generic authorizations such as all (`all`), all administration (`alladm`), all MQI (`allmqi`), and no authorizations (none)

IBM Training

Set or reset authorization

- Use **setmqaut** command to set or reset authorization by object type
  - Prefix authorization with a plus sign (+) to add
  - Prefix authorization with a minus sign (-) to revoke
- Command is cumulative
  - Option 1: Set authorization explicitly on each **setmqaut** command to avoid retaining unwanted pre-existing authorities
  - Option 2: Specify **-all** to remove all authorization and then grant required authorizations

Format:

```
setmqaut -m QMgr -t Objtype -n Profile
[-p Principal | -g Group] permissions
```

Example:

```
setmqaut -m JUPITER -t queue -n MOON.* -g VOYAGER +browse -put

setmqaut -m QM -t queue -n Q1
-p cn=useradm,ou=users,o=ibm,c=UK +put
```


Figure 10-10. Set or reset authorization

Three control commands provide control and verification of the security environment for MQ: **setmqaut**, **dspmqaut**, and **dmpmqaut**. These programs require a connection to the authorization service; they can be used when the target queue manager is active and the OAM is enabled. All of the responses to these commands are displayed on the screen.

The **setmqaut** command sets the access to a particular resource by a principal or group. This command can be used to add or revoke privileges.

The **setmqaut** command can use generic profiles. In the first example, the **setmqaut** command allows members of the group VOYAGER (`-g VOYAGER`) to browse (`+browse`) but not put (`-put`) messages on the queues (`-queues`) that start with the characters “MOON” (`-n MOON.*`) that the JUPITER queue manager (`-m JUPITER`) owns.

The second example shows how to use the LDAP form to reference a principal in an LDAP repository (`-p`). This command sets put permissions (`+put`) on the queue (`-queue`) that is named Q1 (`-n Q1`) on the queue manager that is named QM (`-m Q1`).

If you use the **setmqaut** command to set authorizations for an individual user ID, the authorizations are held at the level of the individual user ID. However, for MQ on UNIX, authorizations are held at the level of the primary group of the user ID, and so all members of that group acquire the same authorizations.

The control command **setmqaut** is used to grant and revoke authorizations to an MQ object. Using the command, you can grant or revoke authorizations for an individual user ID or for a group.

Display authorization

- Use **dspmqaut** command to verify that object authorization is correct
  - By object type
  - For a principal or group

Format:

```
dspmqaut -m QMgr -t ObjType -n ObjName
 [-p Principal | -g Group] [-s Service]
```

Example:

```
dspmqaut -m SATURN -t q -n APPL.Q1 -p mquser
Entity mquser has the following authorizations for object
APPL.Q1:
  get
  browse
  ...
```


Figure 10-11. Display authorization

The **dspmqaut** command displays current authorizations for MQ objects that include queues, queue managers, and processes.

This command does support generic profiles. If a user ID is a member of one or more groups, the display authorization command displays the combined authorizations of all the groups.

Only one group or principal can be specified.

The example command displays the authorizations for the queue (`-q`) that is named APPL.Q1 (`-n APPL.Q1`) on queue manager SATURN (`-m SATURN`) for the user that is named “mquser” (`-p mquser`).

Create authorization report

- Use **dmpmqaut** command to generate a report of current authorizations
  - By object type
  - For a principal or group

Format:

```
dmpmqaut -m Qmgr -t Objtype [-n Profile | -l]
      [-p Principal | -g Group]
```

Example:

```
dmpmqaut -m qm1 -t q -n a.b.c -p user1
profile: a.b.*
object type: queue
entity: user1
type: principal
authority: get, browse, put, inq
```


Figure 10-12. Create authorization report

Use the MQ **dmpmqaut** control command to create a report of the current authorizations that are associated with a specified profile (-n).

The **-l** parameter creates a report with a terse list of all profiles names and types.

Group names must exist, and you can specify one group name on the **dmpmqaut** command. MQ for Windows allows the use of local groups only.

The example in the figure creates a report that shows the object authority for the queues on queue manager **qm1** for the user **user1**.

IBM Training

Using MQSC to manage authorization

- Use MQSC command **SET AUTHREC** to set authority records that are associated with a profile name

  Example:

```
SET AUTHREC OBJTYPE(QMGR)
PRINCIPAL('JohnDoe1@yourcompany.com') AUTHADD(connect)
```

- Use MQSC command **DISPLAY AUTHREC** to display authority records that associated with a profile name

- Use MQSC command **DELETE AUTHREC** to delete authority records

You must use MQSC to manage authorization on the IBM MQ Appliance

Figure 10-13. Using MQSC to manage authorization

You can use the MQSC **SET AUTHREC**, **DISPLAY AUTHREC**, and **DELETE AUTHREC** commands to manage authorizations on a specific queue manager.

The list of authorizations to add and the list of authorizations to remove must not overlap. For example, you cannot add “display” authority and “remove display” authority with the same command. This rule applies even if the authorities are expressed by using different options.

For example, the following command fails because DSP authority overlaps with ALLADM authority:

```
SET AUTHREC PROFILE(*) OBJTYPE(QUEUE) PRINCIPAL(PRINC01) AUTHADD(DSP)
AUTHRMV(ALLADM)
```

IBM Training

Using IBM MQ Explorer to manage authorization

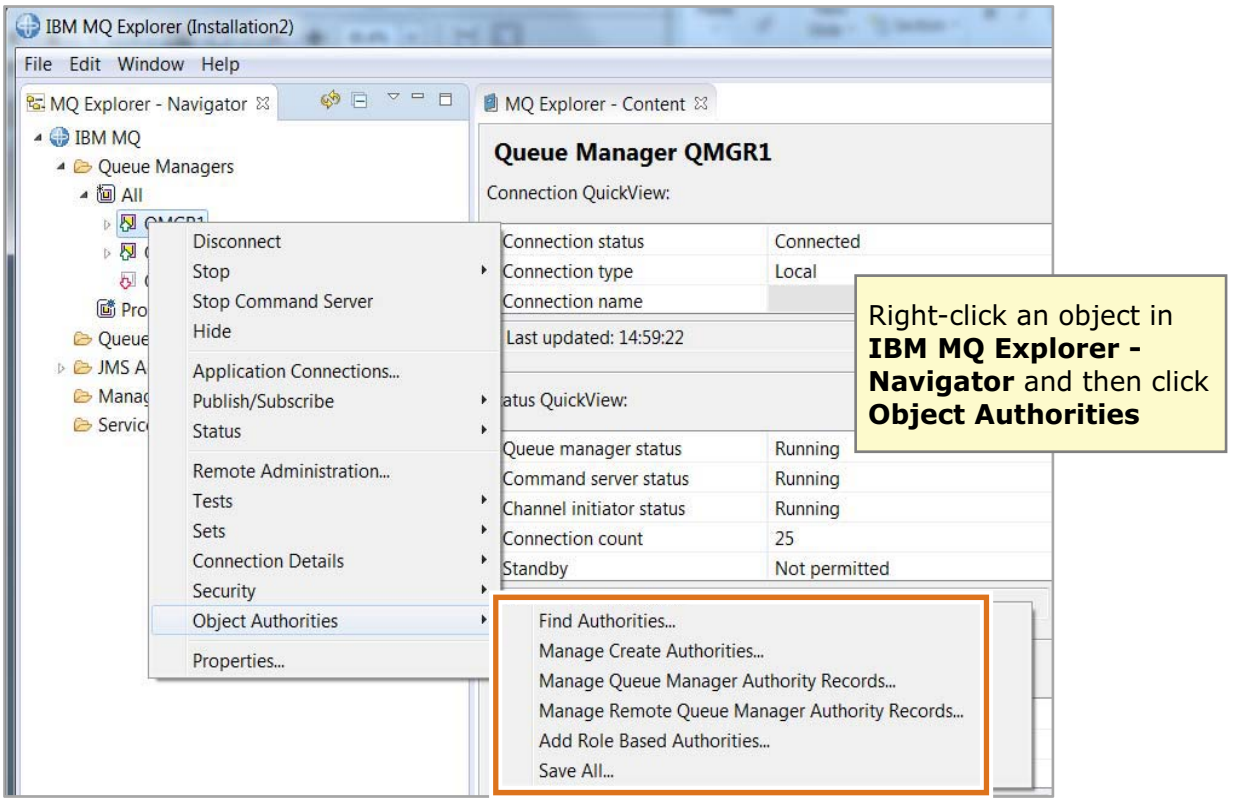


Figure 10-14. Using IBM MQ Explorer to manage authorization

As an option, you can manage object authorities with MQ Explorer.

In MQ Explorer, you can:

- Find authorities
- Manage “create” authorities
- Manage queue manager authority records
- Manage remote queue manager authority records
- Add role-based authorities

For example, to manage the authority records for a queue manager, right-click the queue manager in the **MQ Explorer - Navigator** view and then click **Object Authorities**.

IBM Training

Queue manager authorization in IBM MQ Explorer

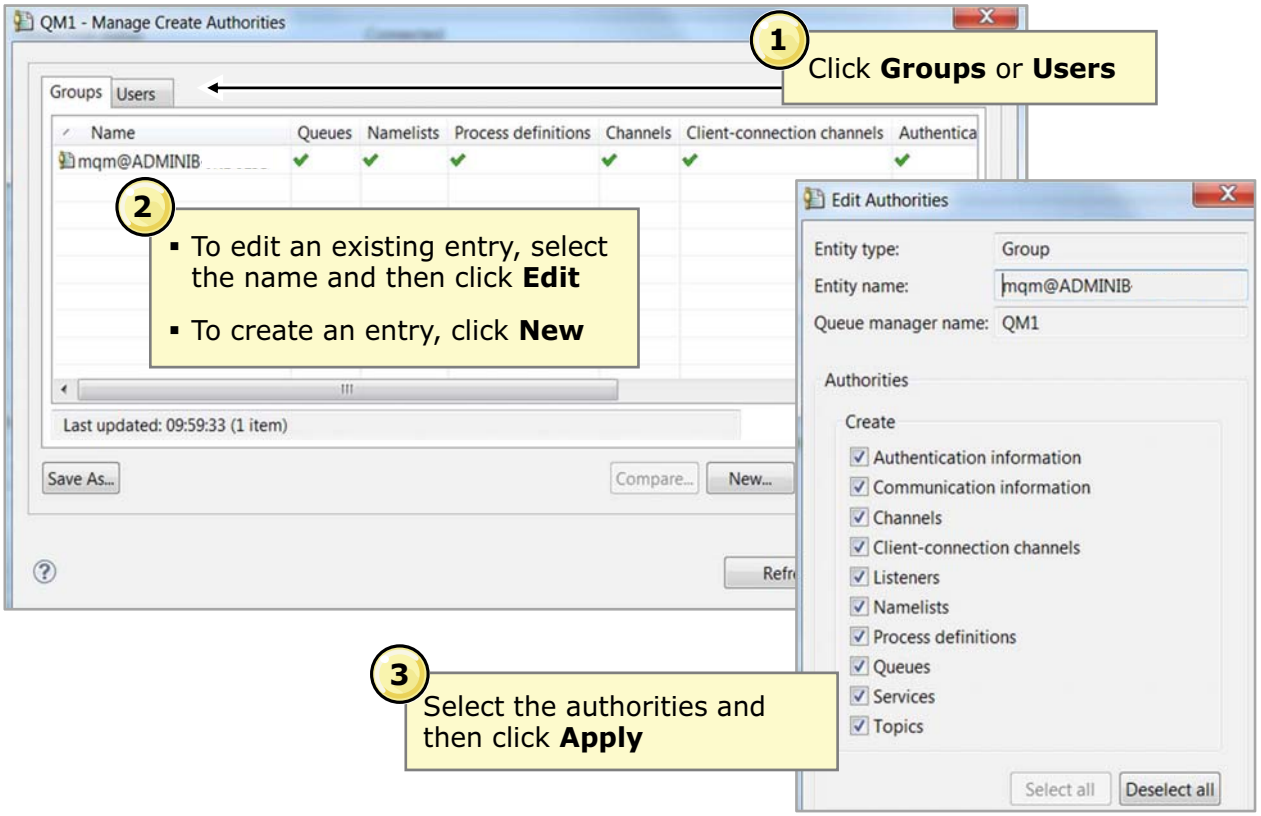


Figure 10-15. Queue manager authorization in IBM MQ Explorer

To modify an existing authority record, complete the following steps:

1. Click the **Groups** tab to modify a group record, or click the **Users** tab to modify the record for a specific user.
2. Select the group or user and then click **Edit**.
3. Modify the authorities and then click **OK**.

To create a new authority record:

1. Click the **Groups** tab to add a record for a group, or click the **Users** tab to add a record for a specific user.
2. Click **New**.
3. Enter an entity name, select the authorities, and then click **OK**.

IBM Training

Queue authorization in IBM MQ Explorer

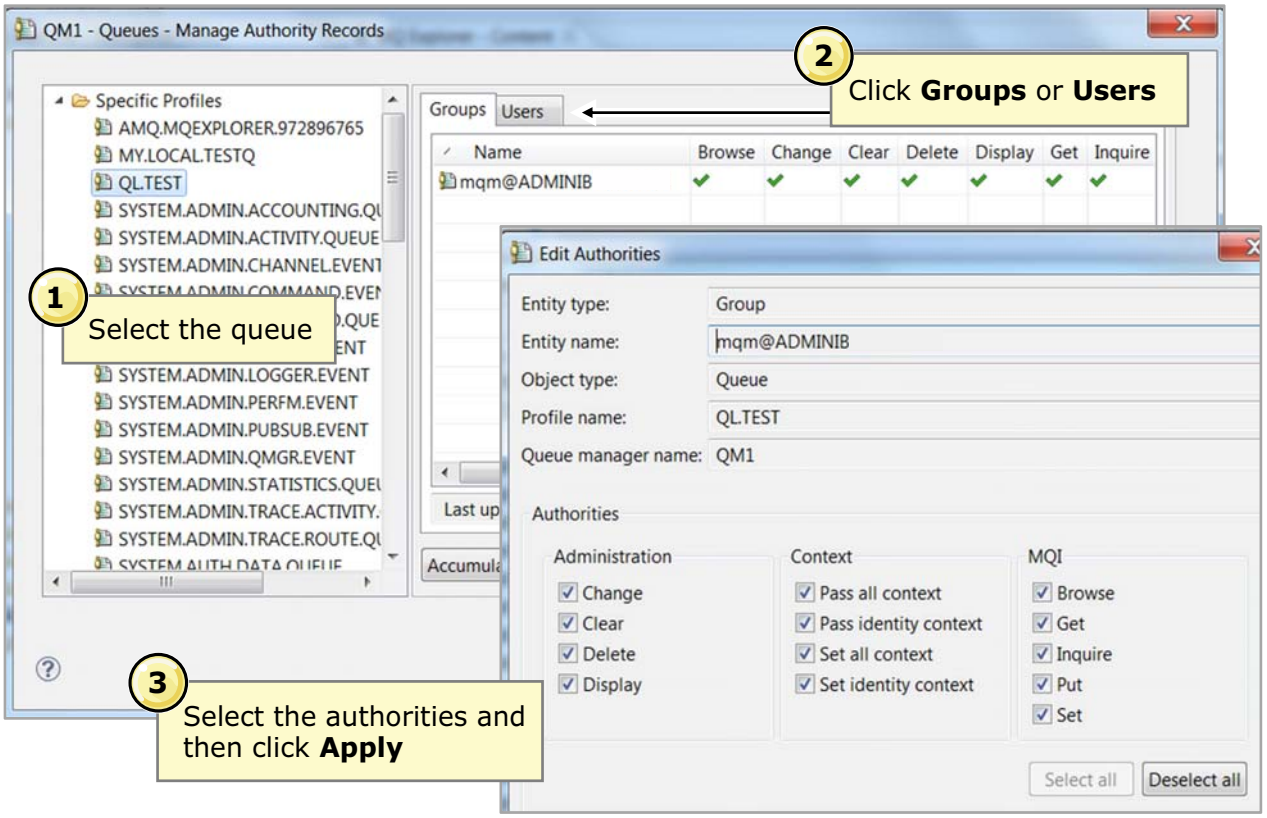


Figure 10-16. Queue authorization in IBM MQ Explorer

You can modify or add queue authority records in MQ Explorer, by right-clicking the queue in the **Queue** content view and then clicking **Object Authorities > Manage Authority Records**.

On the **Manage Authority Records** view, complete the following steps:

1. Select the queue.
2. Click the **Groups** or **Users** tab. Select the record and then click **Edit** to modify an existing record, or click **Add** to add a record.
3. Select the authorities and then click **Apply**.

IBM Training

Channel authorization in IBM MQ Explorer

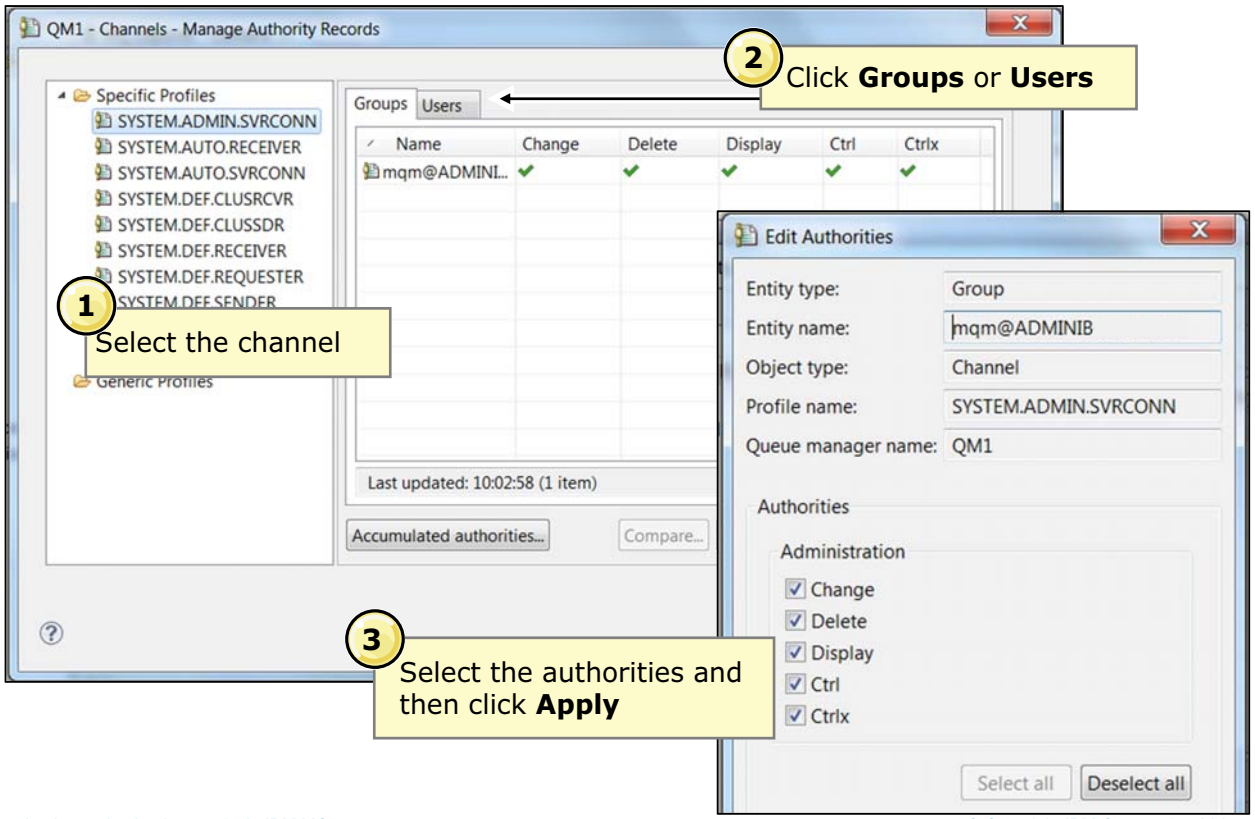


Figure 10-17. Channel authorization in IBM MQ Explorer

You can also use MQ Explorer to add or modify channel authority records by using the same technique that is used to add or modify queue authority records.

Right-click the channel in the **Channels** content view and then click **Object Authorities > Manage Authority Records**.

On the **Manage Authority Records** view, complete the following steps:

1. Select the channel.
2. Click the **Groups** or **Users** tab. Select the record and then click **Edit** to modify an existing record, or click **Add** to add a record.
3. Select the authorities and then click **Apply**.

IBM Training

Access control for IBM MQ control commands

- Use of most IBM MQ control commands is restricted
  Example: **`crtmqm`**, **`strmqm`**, **`runmqsc`**, **`setmqaut`**
- UNIX and Linux restricts users to `mqm` group
  - Configuration as a part of IBM MQ installation
  - Control that the operating system imposes, not OAM
- Windows allows:
  - `mqm` group
  - `Administrators` group
  - System user ID

Figure 10-18. Access control for IBM MQ control commands

Access control can be configured for most MQ control programs. Control programs include commands for creating queue managers, starting queue managers, running MQSC, and setting authorization.

By default, UNIX and Linux restrict access to control programs to members of the “mqm” group.

By default, Windows restricts access to control programs to members of the “mqm” and “Administrators” groups, and the Windows System ID.

IBM Training

Security and distributed queuing

- Put authority option for receiving end of message channel
  - Default user identifier is used
  - Context user identifier is used
- Transmission queue
  - Messages that are destined for remote queue manager are put on transmission queue by local queue manager
  - Application does not normally need to put messages directly on transmission queue, or need authorization to do so
  - Only special system programs should put messages directly on a transmission queue and have the authorization to do so

Figure 10-19. Security and distributed queuing

You can also implement security between queue managers. On the receiving end of a message channel, you can specify the user ID to use for checking the authority of the receiving MCA to open a destination queue.

- Choose **Default user identifier** to use the receiving MCA's default user identifier. A security exit or setting the MCAUSER attribute in the channel definition at the receiving end of the message channel can change this user identifier.
- Choose **Context user identifier** to use the user identifier in the context of the message.

In a typical implementation, it is the queue manager that puts messages that are destined for a remote queue manager onto the transmission queue. For special cases and custom applications, you can allow special system programs only to put messages directly on a transmission queue.

IBM Training

Security authorization for remote queues

- Distributed operating systems have authorizations for remote and clustered queues
- For applications that explicitly open **queue@qmgr**, which is a common pattern when using reply to information

Example: `setmqaut -m QM1 -t rqmname -n QM2 -p mquser +put`

Figure 10-20. Security authorization for remote queues

MQ supports security authorization for remote and clustered queues by using a remote queue manager object that the OAM recognizes. Authorities are applied to the remote queue manager object instead of the transmission queue that is used to send the message to the remote queue.

An example of a command to set security authorization for a non-local queue is provided in the figure. The example gives the user “mquser” put authority on the remote queue (`-t rqname`) that is named QM2 (`-n QM2`).

When an application is attempting to access an MQ object, the general rule is that its authority is checked on the first object in the resolution path. For example, if the object descriptor supplies the name of a remote queue and remote queue manager, authority checking occurs on the transmission queue with the same name as the remote queue manager. If the application attempts to open a local definition of a remote queue, authority checking occurs against that object. For an alias queue, authority checking occurs at the level of the alias queue, not at the level of the queue to which it resolves.

Limit the ability to define queues to privileged users. Otherwise, normal access control can be bypassed by creating an alias queue. The use of the **+crt** authorization on the **setmqaut** control command allows you to specify which users are allowed to create queues.

Example: **setmqaut -m QMC01 -t queue -g GROUPB +crt**

IBM Training

Authorization checking in the MQI

- MQI calls with security checking
  - `MQCONN` and `MQCONNX`
  - `MQOPEN`
  - `MQPUT1` (implicit `MQOPEN`)
  - `MQSUB`
  - `MQCLOSE` (for dynamic queues)

- If not authorized, then reason code **`MQRC_NOT_AUTHORIZED`** is returned

Figure 10-21. Authorization checking in the MQI

This figure lists the MQI calls with security checking.

Applications that send the MQCONN or MQCONNX call must be authorized to connect to the queue manager.

For MQOPEN and MQPUT1, the authority check is made on the name of the object that is opened, and not on the name or names that result after a name is resolved. For example, an application might be granted authority to open an alias queue without having authority to open the base queue to which the alias resolves. The rule is to check the first definition that is encountered during the process of resolving a name that is not a queue manager alias, unless the queue manager alias definition is opened directly.

When an MQSUB call is sent, the queue manager verifies that the user identifier under which the application is running has the appropriate level of authority to subscribe to the topic object.

The MQCLOSE call has options for deleting and purging permanent dynamic queues. If one of these options is set, a check determines whether the user is authorized to delete the queue. This check is not done if the application that created the queue is attempting to delete the queue.

If authorization fails for any of the MQI calls, an MQRC_NOT_AUTHORIZED event is written to the SYSTEM.ADMIN.QMGR.EVENT event queue. This event indicates that a command is entered by a user ID that is not authorized to access the object that is specified in the command.

Authority events (1 of 2)

- Queue manager event
- Stored on SYSTEM.ADMIN.QMGR.EVENT queue

- MQRC_NOT_AUTHORIZED event types
  1. On MQCONN or system connection call, the user is not authorized to connect to the queue manager
  2. On an MQOPEN or MQPUT1, the user is not authorized to open the object for the options specified
  3. When closing a queue with MQCLOSE, the user is not authorized to delete the object, which is a permanent dynamic queue
  4. Command was issued from a user ID that is not authorized to access the object specified in the command
  5. On an MQSUB, the user is not authorized to subscribe to the specified topic
  6. On an MQSUB, the user is not authorized to use the destination queue with the required level of access

Figure 10-22. Authority events (1 of 2)

Authority events report an authorization, such as an application that tries to open a queue for which it does not have the required authority. It might also be a command that is issued from a user ID that does not have the required authority.

Authority events (2 of 2)

- Enable by setting **AUTHOREV(ENABLED)** on queue manager or **Events > Authority events** queue manager property to **Enabled** in IBM MQ Explorer

- Use the **amqsevt** sample program to format and display queue manager events

  Example:

```
amqsevt -m QM01 -q SYSTEM.ADMIN.QMGR.EVENT
    Event Type              : Queue Mgr Event [44]
    Reason                  : Not Authorized [2035]
    Event created           : 2016/10/26 09:52_04.54 GMT
    Queue Mgr Name          : QM01
    Reason Qualifier        : Conn Not Authorized
    User Identifier         : oamlabuser
    Appl Type               : Unix
    Appl Name               : amqsput
```


Figure 10-23. Authority events (2 of 2)

You can enable authority events by modifying the queue manager properties with the **ALTER QMGR** command or by using IBM MQ Explorer.

Channel authentication control

- Enabled by default
- Rules are based on:
  - Connecting IP address
  - Connecting queue manager name
  - SSL distinguished name
  - Asserted identity (including *MQADMIN option)
  - Derived identity from distinguished name mapping
- Rules can be applied in WARNING mode to allow connection but generate errors

Figure 10-24. Channel authentication control

In MQ, channel authorization (CHLAUTH) records define the rules that are applied when a queue manager or client attempts to connect through a channel. Channel authorization is enabled by default.

Channel authentication uses rules to control access to a channel. A wildcard can be used with the MQ administrator ID (*MQADMIN) in these rules to cover the use of any ID that would otherwise gain automatic administrative rights over a queue manager. Having a generic user ID, such as *MQADMIN, makes it easier to have the same rules on all operating systems, where the actual definition of who is an administrator might vary.

Rules can also be defined as “WARN” and generate authorization events without blocking the connection. This behavior might help when you change to a secure environment by not turning off connections immediately.

Channel access control is taught in detail in course WM113/ZM113, *IBM MQ V9 Advanced System Administration*.

IBM Training

Channel authentication commands

- Use MQSC command **SET CHLAUTH** to create or modify a channel authentication record
- Use MQSC command **DISPLAY CHLAUTH** to test rules that you define
- Use MQSC command **ALTER QMGR CHLAUTH(DISABLED)** to disable channel authentication

Figure 10-25. Channel authentication commands

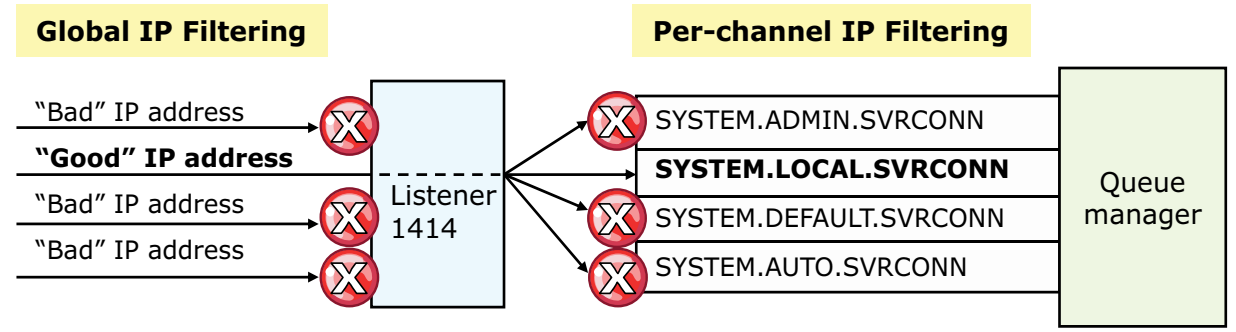
Use the MQSC command **SET CHLAUTH** to configure channel authentication control.

Use the MQSC command **DISPLAY CHLAUTH** with the **MATCH(RUNCHECK)** option to verify a simulated connection. Rules can be tested from the console without making a real connection.

If necessary, such as in a development environment, you can disable channel authentication by using the MQSC command **ALTER QMGR CHLAUTH(DISABLED)**.

IBM Training

Channel authentication example




```
SET CHLAUTH('*') +
 TYPE(BLOCKADDR) +
 ADDRLIST('192.168.1.*')
```

```
SET CHLAUTH('SYSTEM.*') +
 TYPE(ADDRESSMAP) ADDRESS ('*') +
 USERSRC(NOACCESS)
SET CHLAUTH('SYSTEM.LOCAL.SVRCONN') +
 TYPE(ADDRESSMAP) ADDRESS('127.0.0.*') +
 USERSRC(CHANNEL)
```

- Filter connection requests based on IP address of requester
- Per-channel rules match least-specific to most-specific
- Global blocking rules occur at listener before channel name is known and take precedence over per-channel rules

Figure 10-26. Channel authentication example

This figure provides an example of channel authentication.

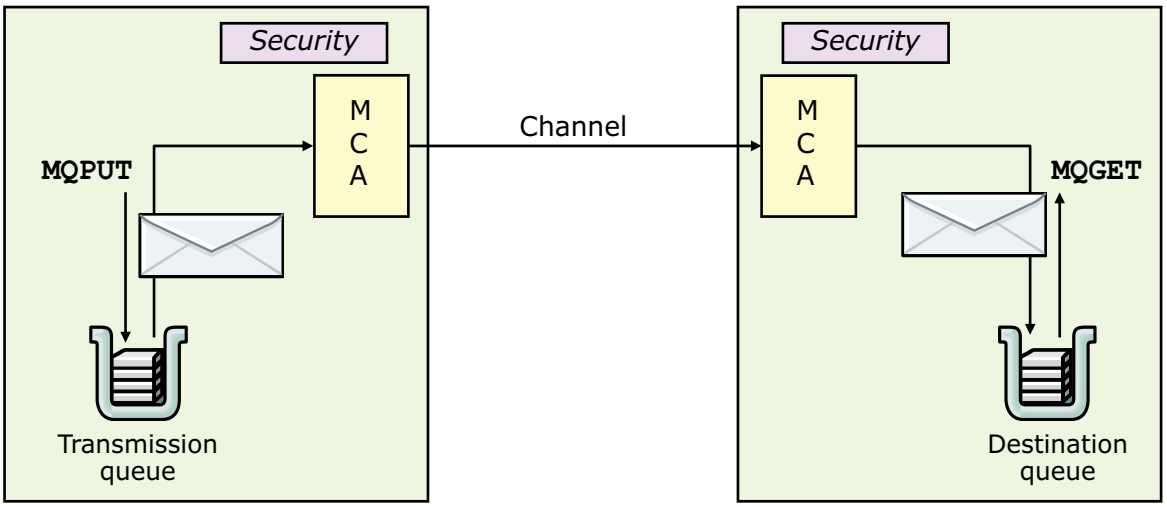
The **SET CHLAUTH** command with the **TYPE(BLOCKADDR)** option provides global IP filtering. In the command, the address list (**ADDRLIST**) is the address from which you are refusing connections. The address can be a specific IP address or a pattern that includes the asterisk (*) as a wildcard or the hyphen (-) to indicate a range that matches the address. In this example in the figure, the **SET CHLAUTH** command blocks all IP addresses beginning with 192.168.1.

In the example on the right, the first **SET CHLAUTH** command restricts access to all SYSTEM channels from any IP address because the **USERSRC** parameter is set to **NOACCESS**. The **NOACCESS** option means that inbound connections that match this mapping do not have access to the queue manager and the channel ends immediately.

The second command gives the local host (127.0.0) access to the channel that MQ Explorer uses because the **USERSRC** parameter is set to **CHANNEL**. The **CHANNEL** option means that inbound connections that match this mapping use the flowed user ID or any user that is defined on the channel object in the **MCAUSER** field.

IBM Training

IBM MQ security exits



- Allows an MCA to authenticate its partner
- Usually work in pairs at each end of channel
- Called immediately after initial data negotiation completes on channel startup, but before any messages start to flow
- Formats of security message and security exit program are user-defined

Figure 10-27. IBM MQ security exits

If the MQ provided security options do not provide the level of support that your application requires, you can use a *channel security exit* to use a custom application for security.

A channel security exit forms a secure connection between two security exit programs, where one program is for the sending MCA, and one is for the receiving MCA.

Security exits normally work in pairs, one at each end of a channel. They are called immediately after the initial data exchange negotiation completes on channel startup, but before any messages start to flow. One possible outcome of the conversation between the security exits is that the channel is closed and message flow is not allowed to proceed.

The name of the security exit is specified as a parameter on the channel definition at each end of a channel.

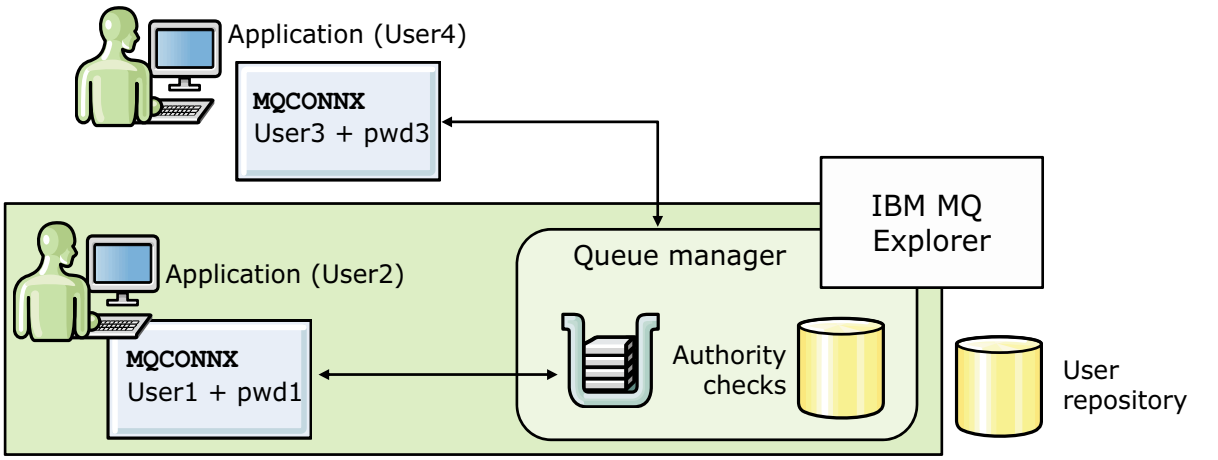
Note

Try to use MQ options such as SSL or connection authentication before you write a custom exit.

IBM Training

Connection authentication

- Client application can provide a user ID and password
- Queue manager can be configured to act on a supplied user ID and password
- LDAP repository can be used to determine whether a user ID and password combination is valid
- Application user ID, which is the usual operating system user ID presented to IBM MQ, might be different from the user ID that the application provides



Implementing basic security in IBM MQ

Figure 10-28. Connection authentication

Connection authentication in MQ can be achieved in various ways:

- An application can provide a user ID and password. The application can either be a client, or it can use local bindings.
- A queue manager can be configured to act on a supplied user ID and password.
- An external repository such as an LDAP repository can be used to determine whether a user ID and password combination is valid.

In the diagram, two applications are making connections with a queue manager, one application as a client and one using local bindings. Applications might use various APIs to connect to the queue manager, but all can provide a user ID and a password. The user ID that the application is running under, User1 and User3 in the diagram, might be different from the user ID that is provided by the application (User2 and User4).

IBM Training

Enabling connection authentication on a queue manager

- Define authentication information object with **DEFINE AUTHINFO** command
  - **IDPWOS** indicates that queue manager uses local operating system to authenticate user ID and password
  - **IDPWLDAP** indicates that queue manager uses an LDAP server to authenticate user ID and password
- Set **CONNAUTH** attribute on queue manager to name of an authentication information object
- Use **REFRESH SECURITY** command to refresh cached view of configuration for connection authentication

Figure 10-29. Enabling connection authentication on a queue manager

To configure a queue manager to use a supplied user ID and password to check whether a user has authority to access resources, enable connection authentication on the queue manager.

First, define an authentication information object. You can define the authentication object to use the local operating system to authenticate the user ID and password (**IDPWOS**). Optionally, you can use an LDAP server to authenticate the user ID and password (**IDPWLDAP**).

After you define the authentication object, set the **CONNAUTH** attribute on the queue manager to the name of an authentication information object.

You must refresh the configuration before the queue manager recognizes the changes.

IBM Training

Enabling connection authentication examples

Example 1: Define an authentication object that uses local file system to authenticate the user ID and password

```
DEFINE AUTHINFO(USE.OS) AUTHTYPE(IDPWOS)
ALTER QMGR CONNAUTH(USE.OS)
REFRESH SECURITY TYPE(CONNAUTH)
```

Example 2: Define an authentication object that uses an LDAP server to authenticate the user ID and password

```
DEFINE AUTHINFO(USE.LDAP) AUTHTYPE(IDPWLDAP) +
CONNAME('ldap1(389),ldap2(389)') +
LDAPUSER('CN=QMGR1') LDAPPWD('passw0rd')
ALTER QMGR CONNAUTH(USE.LDAP)
REFRESH SECURITY TYPE(CONNAUTH)
```


Figure 10-30. Enabling connection authentication examples

This figure provides two connection authentication examples.

Example 1

- The **DEFINE AUTHINFO** command defines an authentication object that is named USE.OS. The **AUTHTYPE(IDPWOS)** attribute directs this authentication object to use the local operating system to authenticate the user ID and password.
- The **ALTER QMGR** command enables connection authorization on the queue manager by using the USE.OS authentication object.
- The **REFRESH SECURITY** command refreshes connection authorization security on the queue manager so that the queue manager recognizes the changes.

Example 2

- The **DEFINE AUTHINFO** command defines an authentication object that is named USE.LDAP that uses an LDAP server to authenticate the user ID and password.
- The **ALTER QMGR** command enables connection authorization on the queue manager by using the USE.LDAP authentication object.
- The **REFRESH SECURITY** command refreshes connection authorization security on the queue manager so that the queue manager recognizes the changes.

IBM Training

Secure Sockets Layer (SSL)

- Protocol that allows transmission of secure data over an insecure network
- Combines these techniques:
  - Symmetric and secret key encryption
  - Asymmetric and public key encryption
  - Digital signature
  - Digital certificates
- Protection
  - Client/server
  - Queue manager and queue manager channels
- Combats security problems
  - Eavesdropping: Encryption techniques
  - Tampering: Digital signature
  - Impersonation: Digital certificates

Figure 10-31. Secure Sockets Layer (SSL)

SSL is an industry-standard protocol for secure communications, involving encryption, authentication, and integrity of data. SSL is supported in both client/server and queue manger/queue manager channels (including clusters). MQ provides many flexible built-in capabilities such as the ability to identify the users based on their fully authenticated identity. This feature removes, for many people, the requirement to set up channel exits, where they were used for security purposes.

SSL uses a combination of public key and symmetric key encryption to ensure message privacy. Before messages are exchanged, an SSL server and SSL client do an electronic handshake during which they agree to use a session key and encryption algorithm. All messages between the client and the server are then encrypted. Encryption ensures that the messages remain private even if eavesdroppers intercept it.

IBM Training

IBM MQ SSL support

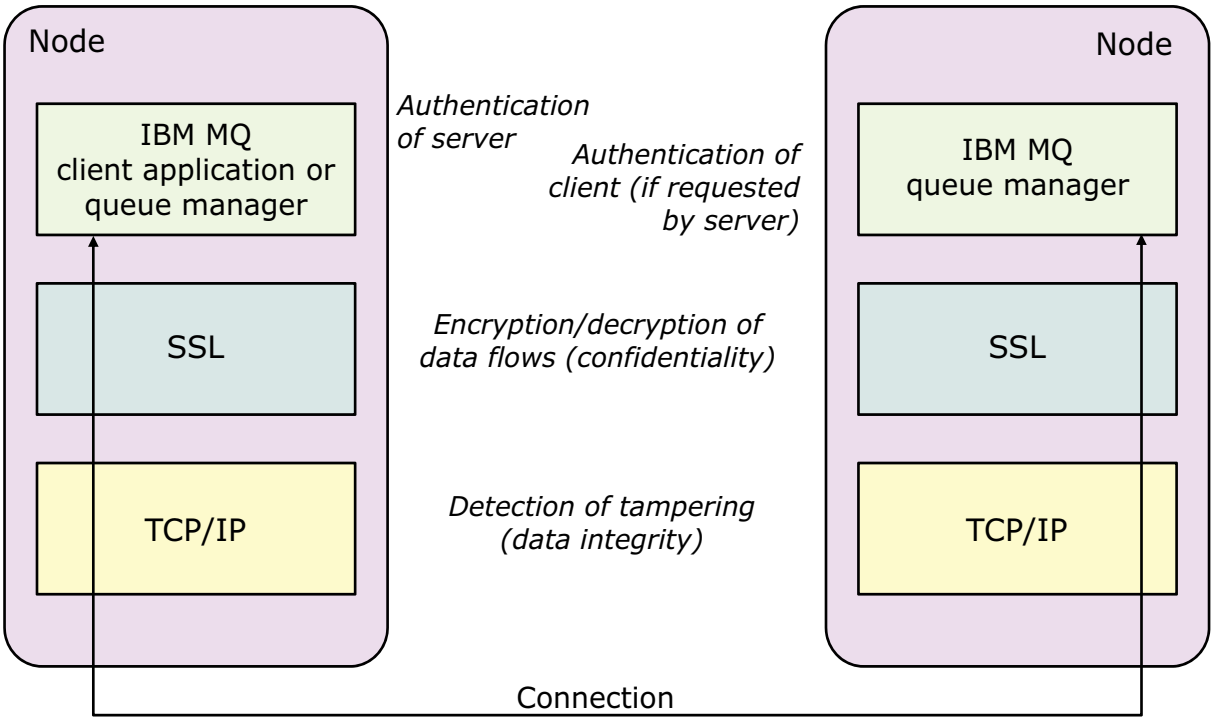


Figure 10-32. IBM MQ SSL support

MQ provides the following support for SSL:

- Authentication of the server on the client
- Authentication of the client, if the server requests it
- Encryption/decryption of data flows
- Detection of tampering

IBM Training

Enabling SSL on the queue manager

- Use IBM MQ Explorer or the MQSC command **ALTER QMGR**

Attribute	Description
SSLCRLNL	Name of a namelist of authentication information objects that provide certificate revocation locations (CRLs) for enhanced TLS/SSL certificate verification
SSLCRYP	Name of parameter string that is required to configure cryptographic hardware present on system
SSLKEYR	Name of SSL key repository
SSLTASKS	Number of server subtasks to use for processing SSL calls
SSLEV	Enable or disable SSL event messages
SSLFIPS	Specify whether only FIPS-certified algorithm is used if cryptography is provided by IBM MQ, rather than cryptographic hardware

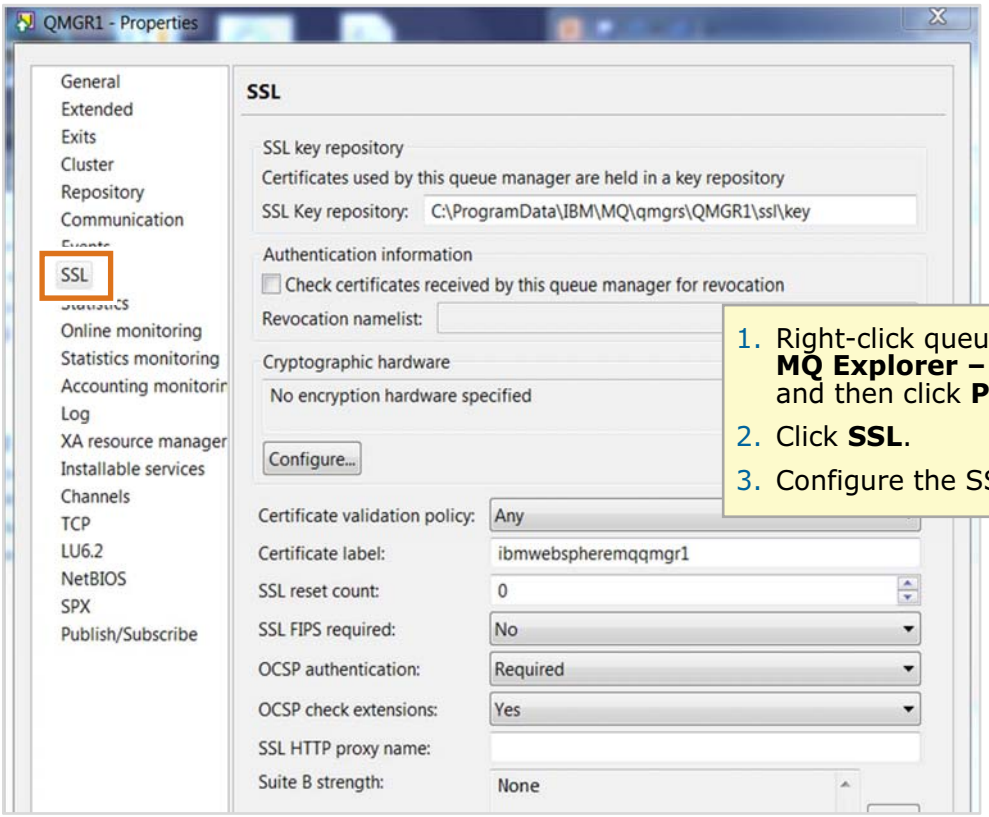
Figure 10-33. Enabling SSL on the queue manager

The figure lists the queue manager attributes that can be modified for SSL by using the **ALTER QMGR** command.

- **SSLCRLNL** (SSL CRL namelist) holds the name for the namelist of authentication information objects.
- **SSLCRYP** (SSL CryptoHardware) holds the name of the parameter string that is required to configure the cryptographic hardware that is present on the system.
- **SSLKEYR** (SSL key repository) holds the name of the SSL key repository.
- **SSLTASKS** (SSL tasks) holds the number of server subtasks to use for processing SSL calls. If you use SSL channels, you must provide two for these tasks.
- **SSLEV** enables or disables the sending of SSL event messages to the channel event queue, when a channel fails to establish an SSL connection.
- **SSLFIPS** specifies whether only FIPS-certified algorithms are used if cryptography is carried out in MQ.

IBM Training

Enabling SSL by using IBM MQ Explorer



1. Right-click queue manager in the **MQ Explorer – Navigator** view and then click **Properties**.
2. Click **SSL**.
3. Configure the SSL properties.

Figure 10-34. Enabling SSL by using IBM MQ Explorer

You can also use MQ Explorer to configure SSL on a queue manager.

IBM Training

Channel attributes for SSL

- Specify cipher specification to use when communicating with the queue manager and ensure that it matches the cipher specification on the target channel
- Use IBM MQ Explorer or the MQSC command **ALTER CHANNEL**

SSLCIPH	Defines a single cipher specification for an SSL connection
SSLPEER	Specifies distinguished name for SSL channel negotiation
SSLCAUTH	Specifies whether channel uses SSL for client authentication

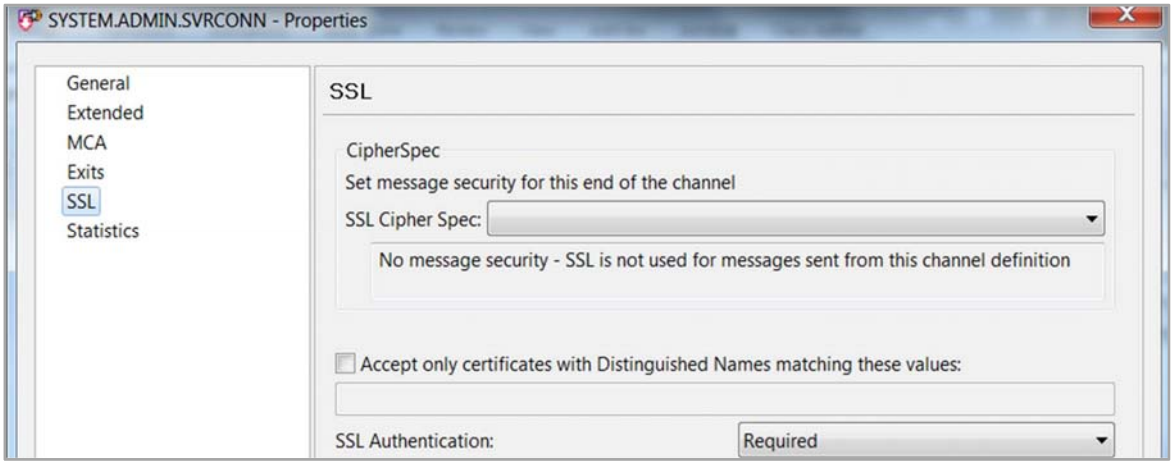


Figure 10-35. Channel attributes for SSL

You can configure the cipher specification to use when you communicate with the queue manager by using the MQSC command **DEFINE CHANNEL** or **ALTER CHANNEL**, or by using MQ Explorer.

The figure lists the attributes available with the **DEFINE CHANNEL** and **ALTER CHANNEL** commands for use with SSL and includes an example of the SSL properties in MQ Explorer.

- **SSLCIPH** specifies the encryption strength and cipher specifications, for example NULL_MD5 or RC4_MD5_US. The cipher specifications must match at both ends of the channel.
- **SSLPEER** specifies the distinguished name (unique identifier) allowed.
- **SSLCAUTH** defines whether MQ requires and validates a certificate from the SSL client.

The server authentication of the client uses the public key of the server to encrypt the data. The server can generate the secret key only if it can decrypt that data with the correct private key.

For client authentication, the server uses the public key in the client certificate to decrypt the data that the client sends during the handshake.

If any step of the authentication fails, the handshake fails and the session ends.

SSL configuration and implementation are described in detail in course WM113/ZM113, *IBM MQ V9 Advanced System Administration*.

Default security configuration in IBM MQ Explorer

- Use **Client Connections** tabs in **Preferences** to set default security options on all new client connections
- Default values can be overridden when a new queue manager is added

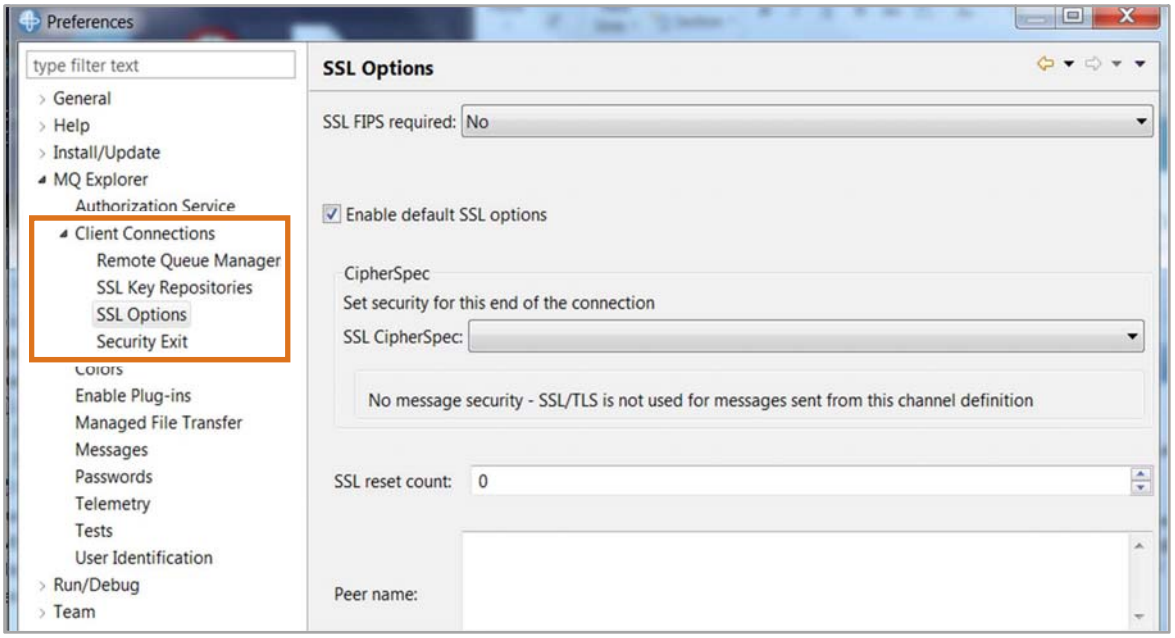


Figure 10-36. Default security configuration in IBM MQ Explorer

Client security configuration support is provided in MQ Explorer. The default security preferences for client connections are part of the **Preferences** dialog box.

To modify the default security preferences for client connections in MQ Explorer, complete the following steps:

1. Click **Windows > Preferences** to open the **Preferences** dialog box.
2. Expand **MQ Explorer**.
3. Expand **Client Connections**. The default security settings for client connections are now accessible.

The figure shows the **SSL Options** client connections preferences.

IBM Training

Storing IBM MQ Explorer passwords

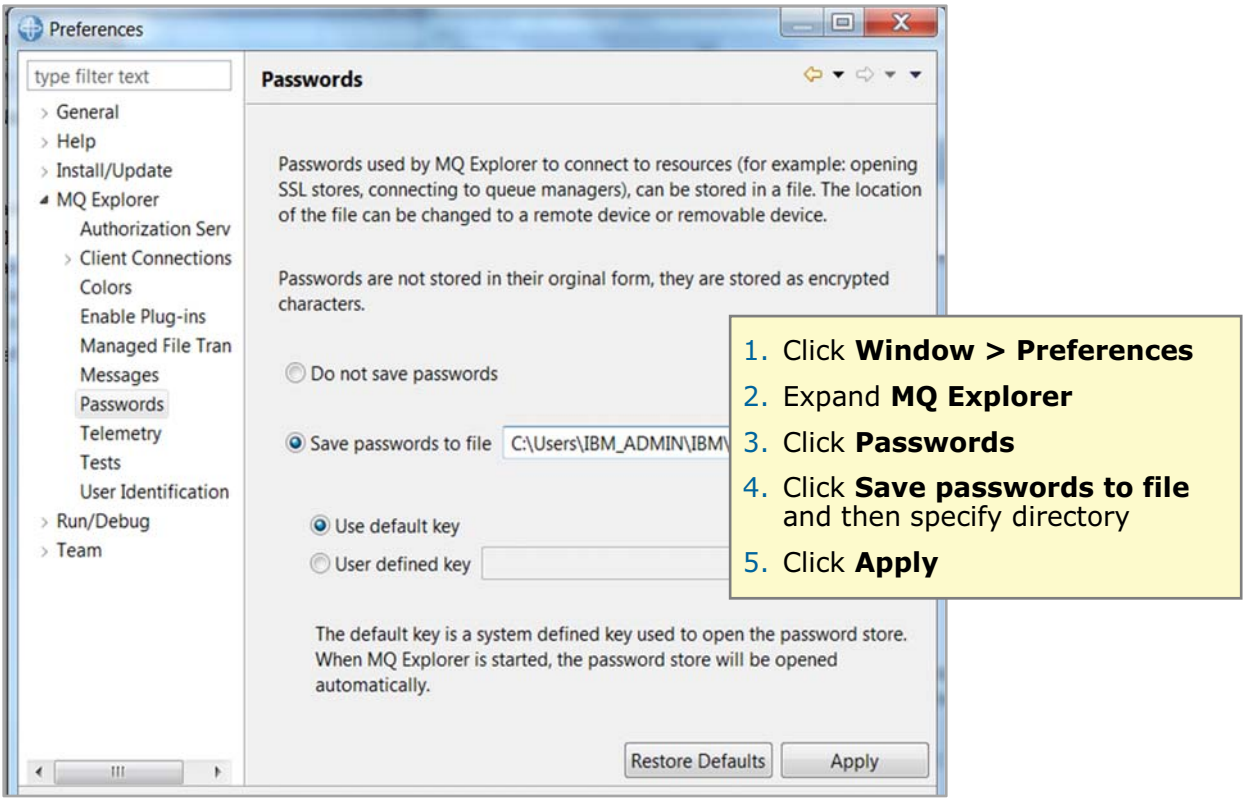


Figure 10-37. Storing IBM MQ Explorer passwords

Using MQ Explorer, you can store passwords to a file so that you do not have to enter them every time you want to connect to resources.

The password file can be stored locally, to a remote device, or to a removable device.

To open the **Passwords** preference window, complete the following steps:

1. Click **Windows > Preferences**.
2. Expand **MQ Explorer**.
3. Select **Passwords** to show the **Passwords** view.

IBM Training

Unit summary

- Describe the role of the object authority manager (OAM) to provide security to IBM MQ resources
- Protect IBM MQ resources by using the OAM
- Use some of the OAM control commands
- Describe the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) support that IBM MQ provides
- Implement basic channel authentication

Figure 10-38. Unit summary

IBM Training

Review questions

1. True or False: You must either restart the queue manager or use the **`REFRESH SECURITY`** command to refresh the cached view of the configuration for connection authentication.

2. For an application to open a queue by using an alternative user ID, the initial user ID requires:
   A. OAM delegate authority
   B. OAM alternate user authority
   C. Password of alternate user ID

3. Using the OAM, which command would you enter to allow PUT access only to the local queue REBATE.IN on MY.QMGR for the group that is named ASSESSORS?
   A. `setmqaut -m MY.QMGR -t q -n REBATE.IN -g ASSESSORS +put`
   B. `setmqaut -m MY.QMGR -t q -n REBATE.IN -p ASSESSORS +put`
   C. `setmqaut -m MY.QMGR -t q -n REBATE.IN -g ASSESSORS -get`

Figure 10-39. Review questions

Write your answers down here:

1.

2.

3.

IBM Training

Review answers

1. True or False: You must either restart the queue manager or use the **REFRESH SECURITY** command to refresh the cached view of the configuration for connection authentication.
   The answer is True.
2. For an application to open a queue by using an alternative user ID, the initial user ID requires:
   A. OAM delegate authority
   B. OAM alternate user authority
   C. Password of alternate user ID
   The answer is B.
3. Using the OAM, which command would you enter to allow PUT access only to the local queue REBATE.IN on MY.QMGR for the group that is named ASSESSORS?
   A. `setmqaut -m MY.QMGR -t q -n REBATE.IN -g ASSESSORS +put`
   B. `setmqaut -m MY.QMGR -t q -n REBATE.IN -p ASSESSORS +put`
   C. `setmqaut -m MY.QMGR -t q -n REBATE.IN -g ASSESSORS -get`
   The answer is A.

Figure 10-40. Checkpoint answers

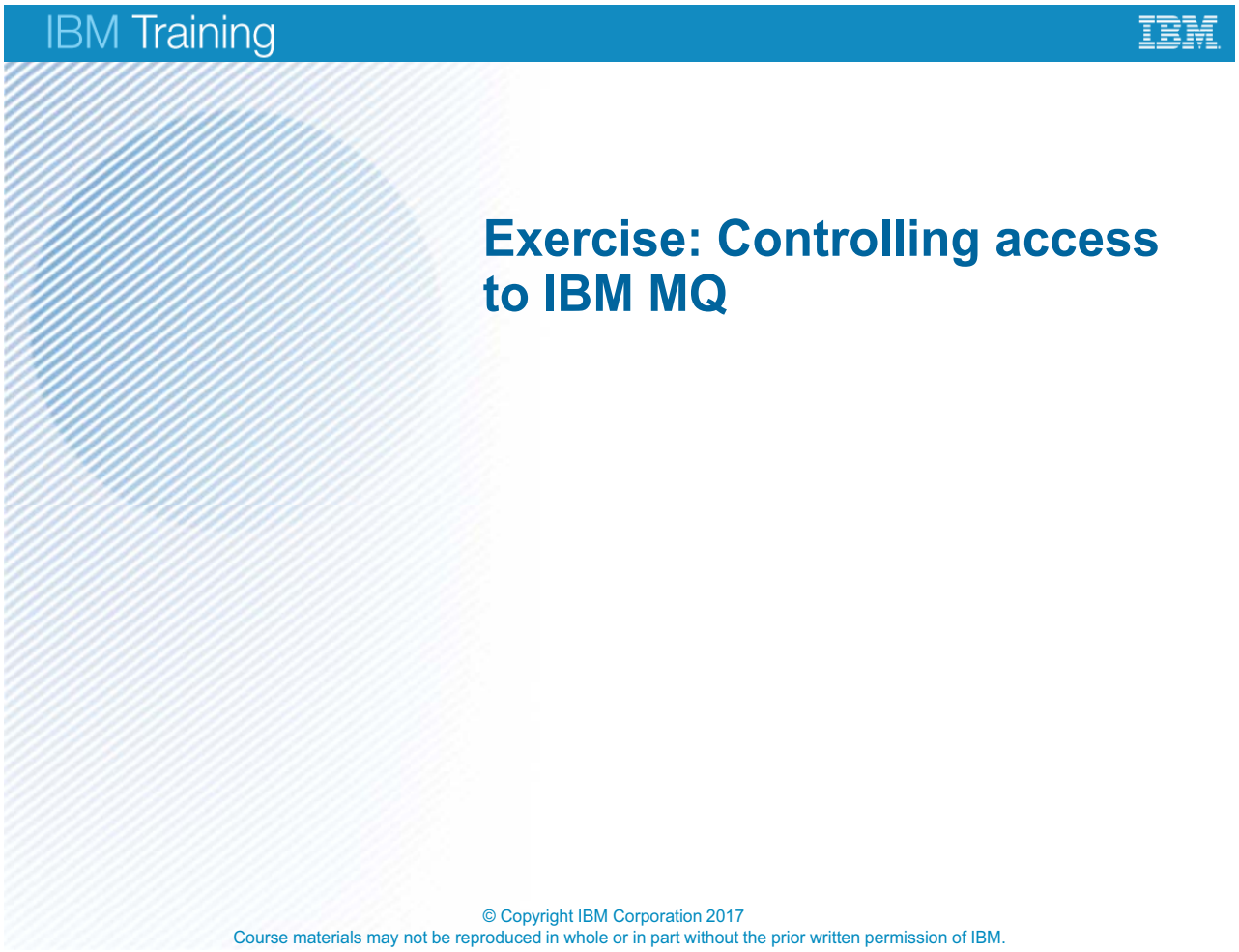


Figure 10-41. Exercise: Controlling access to IBM MQ

In this exercise, you use the OAM utilities to set access control on a queue, and then use sample programs to see the effect of attempting to breach security.

IBM Training

Exercise objectives

- Use the **setmqaut** command to define access control on a queue
- Use the **dspmqaut** command to display the access control on a queue
- Use IBM MQ Explorer to manage authority records
- Enable and monitor authority events
- Test security by using IBM MQ sample programs

Figure 10-42. Exercise objectives

See the *Course Exercises Guide* for detailed instructions.