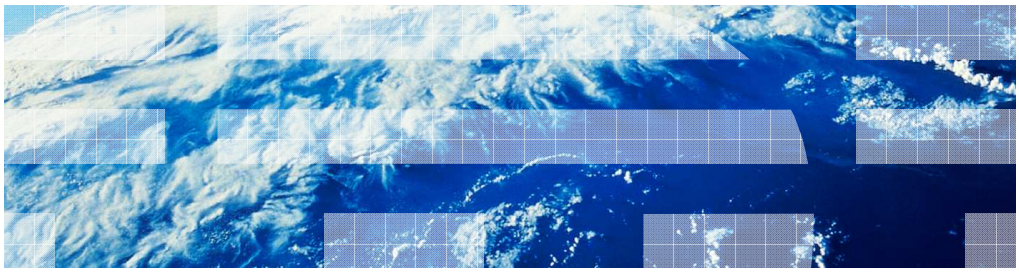
WebSphere Message Broker Version 7

Broker security administration



WebSphere. software

This presentation describes the security administration tools in WebSphere® Message Broker version 7. In earlier versions of Message Broker, the configuration manager was used to hold the access control lists and security mechanisms for Message Broker users. The configuration manager has been removed in version 7, so the security administration function has been updated to reflect this change.

Table of contents

This session will provide a reminder of how security administration is performed in earlier versions of Message Broker, and then explain how the same function has been implemented in version 7. It will show how security is implemented and how end-users will experience the security settings.

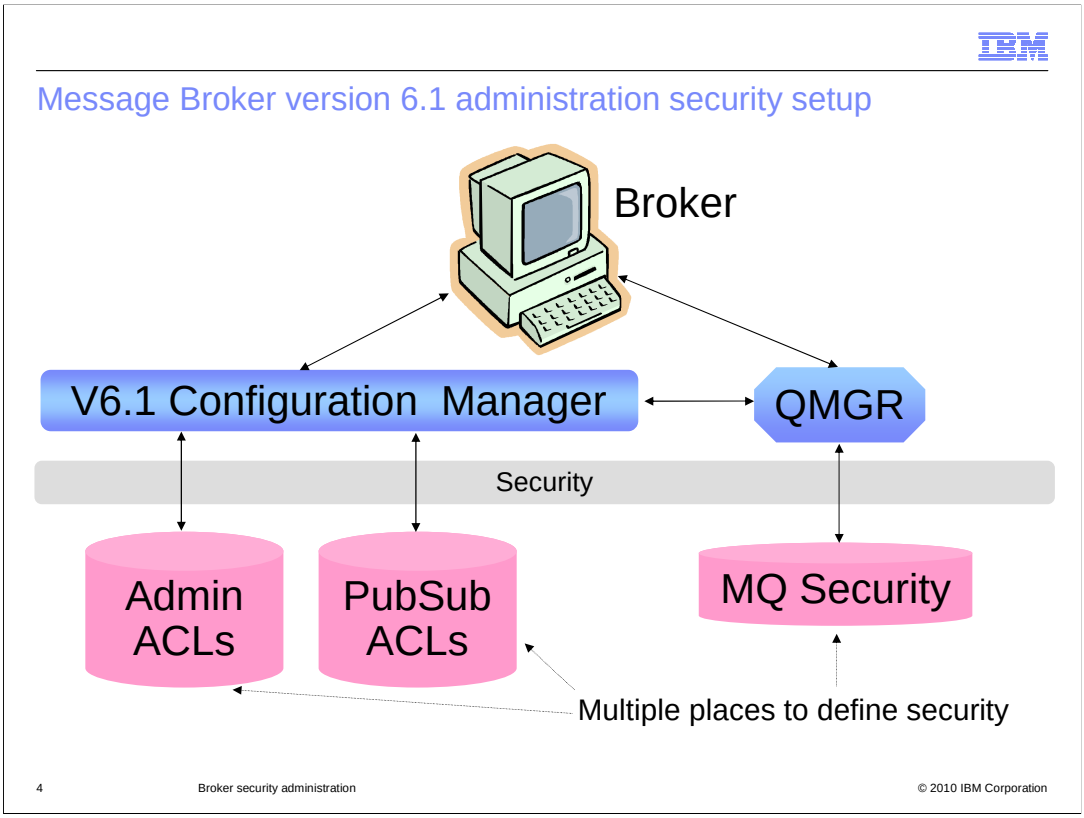
Overview

- Simplification of security administration
- Utilizes MQ security
- Security based on user ID in MQMD
- Security off by default, can be turned on with mqsicreatebroker or mqsichangebroker
- Includes users connecting from:
  - MB Explorer
  - Toolkit
  - CMP API Exerciser
  - CMP API Java™ applications
  - Commands

Message Broker version 7 has simplified the implementation of administration security by making use of MQ as the underlying security policy definition point. All security definitions are made and stored in MQ, and no security information is stored in any Message Broker component. The Message Broker components are only used as security enforcement points, based on security policies stored within MQ.

Security enforcement is based on the user ID that is present in the MQMD header of MQ messages. This then allows you to use the “MCA User” fields in the MQ message to enhance security through MQ channel security.

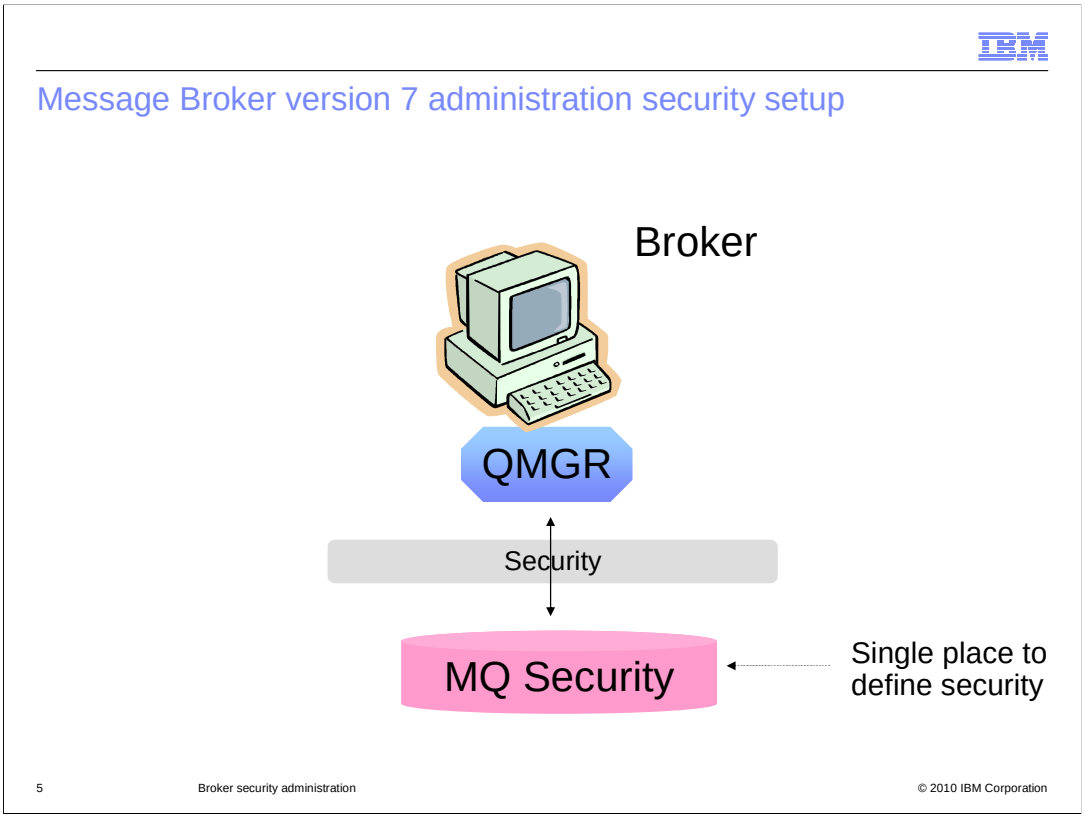
Security is switched off by default, so a new broker will not be activated with security. This can be activated subsequently by using the appropriate command, or the broker creation can be specified to activate security when the broker is created. If security is not enabled, any user can perform any function on the broker, such as create or delete and execution group. Security is applied to any tools that are used to interact with the broker, including the MB Explorer, the Message Broker Toolkit and the command interface.



As a reminder, earlier versions of Message Broker implemented security administration using the configuration manager. The broker runtime component has connectivity to the underlying MQ queue manager, and to the configuration manager.

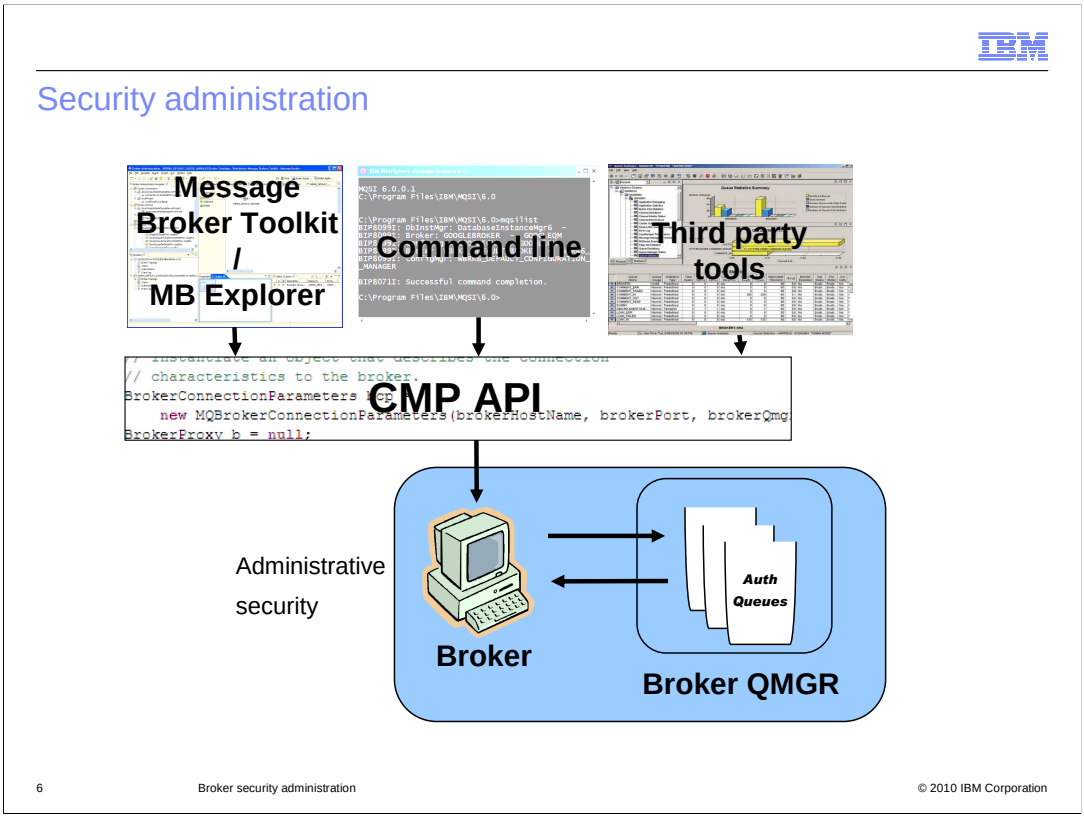
The configuration manager is the primary owner of security access control lists. These apply to both the administration security, and to publish/subscribe users and applications for broker runtime usage. These access control lists are stored in the internal databases managed by the configuration manager.

All these security definition points are quite separate and require different processes to define and manage the appropriate security.



In version 7, the security management is much simpler, with just one place where security is defined and manager. All security administration is now based on MQ, and the tools provided by the queue manager are also used to manage the security for the broker.

Since all publish/subscribe applications are now managed through MQ, this is also used to manage the security requirements for publish/subscribe.



All applications that require access to the broker runtime use the same interface to do this. This is known as the configuration manager application programming interface, or CMP API. All components of the Message Broker product use this interface; additionally, you can write you own applications for administration, and these will use the CMP API to access the broker.

When you use any of these applications, or CMP API, the broker will check your security credentials, and will ensure that you are is permitted to do the intended function. It will do this by asking the queue manager to check the broker security access rules held within the queue manager. These rules, or permissions, are held on a series of MQ queues, known as “authorization queues”. If you need to change these permissions, this can be done by accessing the queue manager directly. A broker restart is not required for this. However, the user’s authority remains until he disconnects from the connected broker. The broker caches security information inside the broker itself, until the user drops his connection. A new user connection user will result in the broker obtaining new security information from the queue manager.

All security administration is performed using native MQ commands, or by using the MQ Explorer. The broker itself does not put or get messages from the AUTH queues. However, the broker requires altuser *(pronounced “alt user”)* authority to check MQ permissions.

Message Broker authorizations

- Message Broker allows three levels of authorization for administrative actions:
  - Reading
  - Writing
  - Executing
- On two object types:
  - Broker
  - Execution group

Message Broker version 7 has simplified the levels of security authorization, and now has just three levels of security. These are read, write and execute, and are generally accepted as a standard security categorization.

These authorities can be applied to two object types, namely the broker itself, and execution groups within the broker.

Authorization queues

- Message Broker permissions are defined with authorization queues
  - **SYSTEM.BROKER.AUTH**
    - One of these per broker
  - **SYSTEM.BROKER.AUTH.<ExecutionGroupName>**
    - One of these per execution group

The broker uses several queues to manage security administration. There is one queue that is used to manage security for the broker itself, and one queue for each of the execution groups that are defined. The MQ queues that are used to manage broker authorizations do not have messages placed on them. They are used purely as a place to define resource authorization, not as a repository for messages.

The broker authorization queue is automatically created when you create a version 7 broker. It is also created when you migrate a version 6 or 6.1 broker to version 7. However, this is only used if security has been activated. If security has not been activated, then this queue will not be used.

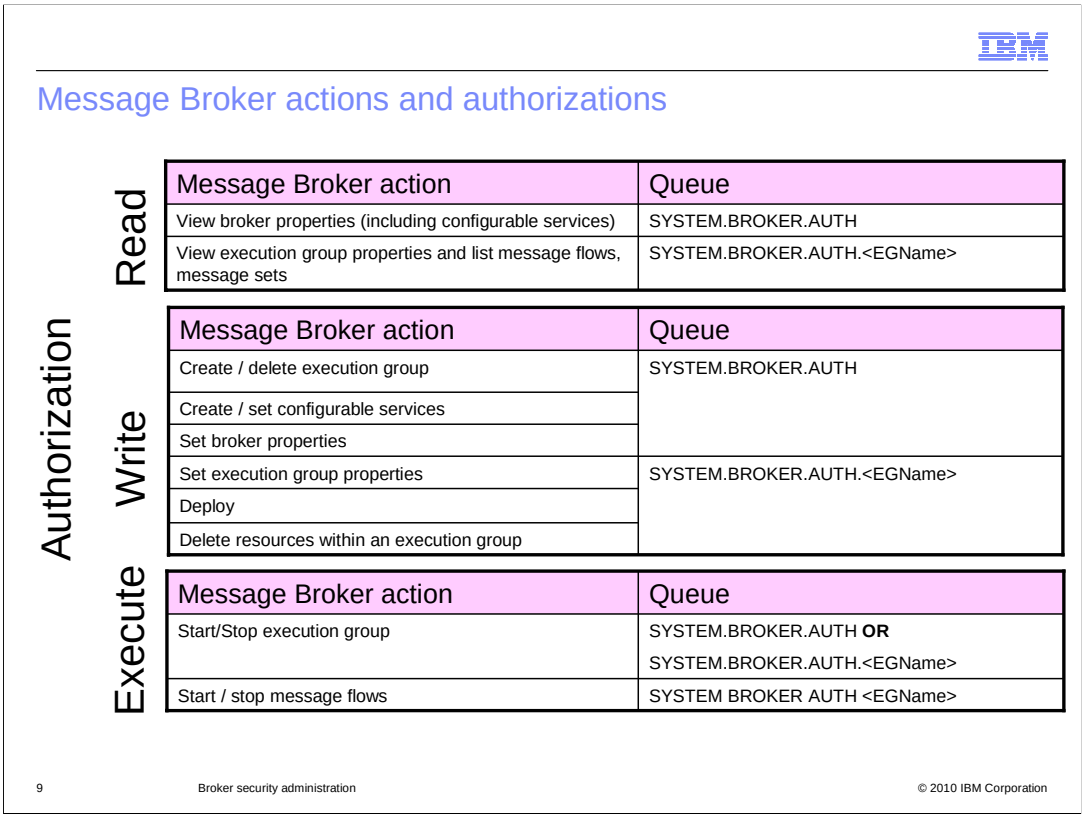
The execution group queue are used to manage security for specific actions against that execution group. The execution group queues will be defined when the execution group is created, but only if security has been activated for the broker.

The default setting is no security. Hence, when a new broker is created, security is not active, and all users are allowed full access to the broker and all execution groups. If you then activate security, or if this is included in the broker creation process, then security is applied in full to all objects, and no access is allowed unless specifically granted. If you activate security on an existing broker, then the process of activating security will automatically create all the required queues for the defined execution groups.

Due to the limitation of an MQ queue name, the execution group name might be truncated.

If two execution groups have similar names, and share the same truncated name, then they will share the same broker auth queue.

MQ queue names have a more restrictive set of characters than an execution group name. Any characters not allowed will be replaced with an underscore. However, both MQ queue names and Message Broker execution groups can both use upper and lower case characters. These will both be honored when matching the authorization queues, and lower case names will not be translated to upper case names.



The read, write and execute authorizations shown on this slide are the primary Message Broker authorizations. This slide shows the broker functions that are contained within these Message Broker authorities.

The “Read” authority contains the Message Broker view actions. These are applicable either to the broker, or to individual execution groups, using the appropriate authorization queue to handle this.

The “Write” authority is used when making updates to the broker system. For example, when creating an execution group, this results in an update action to the broker, so “Write” authority will be required for the broker. When setting the properties of an execution group, this results in an update to the execution group itself, so will, require “Write” authority for the execution group, and the corresponding queue.

The “Write” authorities are the ones that are used when message flow deployment is required, so this is where you will control the main application development and deployment functions. Note that if you deploy a new message flow to an execution group, then this message flow will start execution immediately. This does not mean that you must be authorized for “execute” access on the specific execution group. However, subsequent start and stop actions against that message flow will require “execute” authority. Similarly, you can remove the message flow from the execution group without specific “execute” authority.

Starting and stopping message flows and execution groups is controlled using the “Execute” authority, defined either against the broker, or against individual execution groups. However, there is no implied authority within an execution group, just because a user has broker authority.

Authorization queues creation and deletion

- Creation of broker
  - When a broker is created (or migrated) this queue will be created:
    - **SYSTEM.BROKER.AUTH**
- Creation of execution group
  - If broker security is activated, the broker will attempt to create this queue or queues:
    - **SYSTEM.BROKER.AUTH.<ExecutionGroupName>**
- Deletion of broker
  - When deleting a broker, an optional flag is available to enable automatic deletion of queues
- Deletion of execution group
  - When deleting an execution group, broker will leave the associated authorization queue (for possible future reuse)

When you create or delete brokers or execution groups, the broker will attempt to create or delete the corresponding authorization queues. The broker authorization queue will always be created when the broker is created. If an existing broker has defined execution groups, then the corresponding queues will be created, but only if security has been activated for the broker. This can be done when the broker is first created, using the “-s” option, or at a later time, using the “mqsi” change properties command.

When brokers or execution groups are deleted, you have the option of retaining the authorization queues for later reuse. In the case of execution groups, this is the default action.

A broker may not have the MQ permission to create and delete authorization queues dynamically. If a broker does not have this authority, a message will be written to the system log. The broker administrator must then either manually create the queue, or run mqsi-change-broker to create the queue.

When an execution group is renamed, a new authorization queue must be created which is associated with the new name, with similar permissions to the old queue.

Setting MQ permissions for Message Broker

MB authority	MQ permission
Read	+inq
Write	+put
Execute	+set

- Set the corresponding MQ permission on a broker auth queue to grant a user the Message Broker permission

The Message Broker authorities of read, write and execute are mapped to the equivalent authorities in MQ. The MQ permissions are inquire, put and set.

Examples

- Set **put** on **SYSTEM.BROKER.AUTH** for group "**admin"**
  - Grants write authority for all users in group "admin" to the broker
- Set **inq** on **SYSTEM.BROKER.**** for group "**dev"**
  - Grants read authority for all users in group "dev" for all execution groups
- Set **set** on **SYSTEM.BROKER.AUTH.default** for group "**dev"**
  - Grants execute authority for all users in group "dev" for the specific execution group called 'default'

This slide show three examples of setting broker security, using the appropriate MQ permissions.

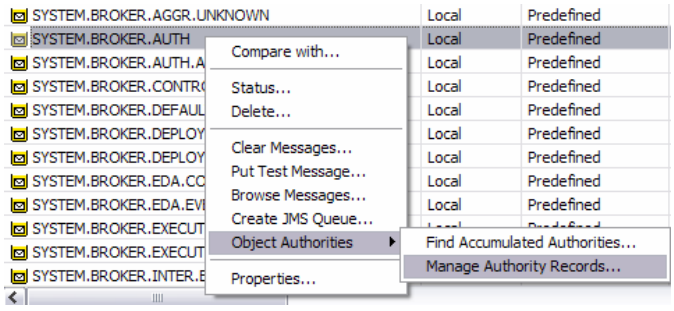
The first example is granting write authority to the entire broker for all users in the "admin" group.

The second example uses the "double asterisk" notation to grant read authority for all users in the "dev" group to all execution groups in the broker. The double asterisk notation means that all execution groups are included, even if the names are nested.

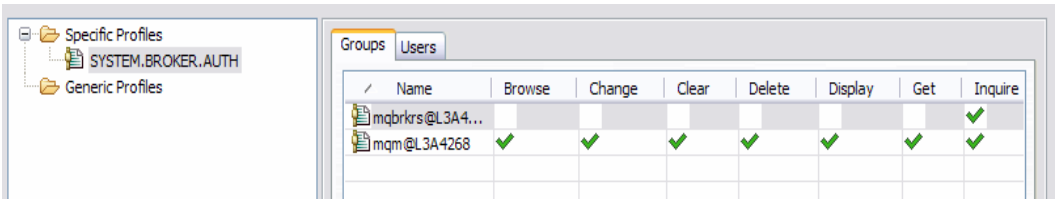
The third example set execute authority for the specific execution group "default".

These permissions can either be set using the "set mq aut" commands, or by using the MQ Explorer, on the next slide.

User interface - MQ Explorer



- MQ Explorer provides graphical user interface to manage Message Broker security



This slide shows the MQ Explorer. The broker's queue manager has been selected, and the queues owned by that queue manager are shown. Select the required authorization queue, right-click, and select "object authorities" from the pop-up menu. Then select "manage authority records", and you will see information similar to the screen capture shown here.

Using this window, you can specify authority for both users and groups. You can also use the MQ tools to search for defined authorities for a particular user.

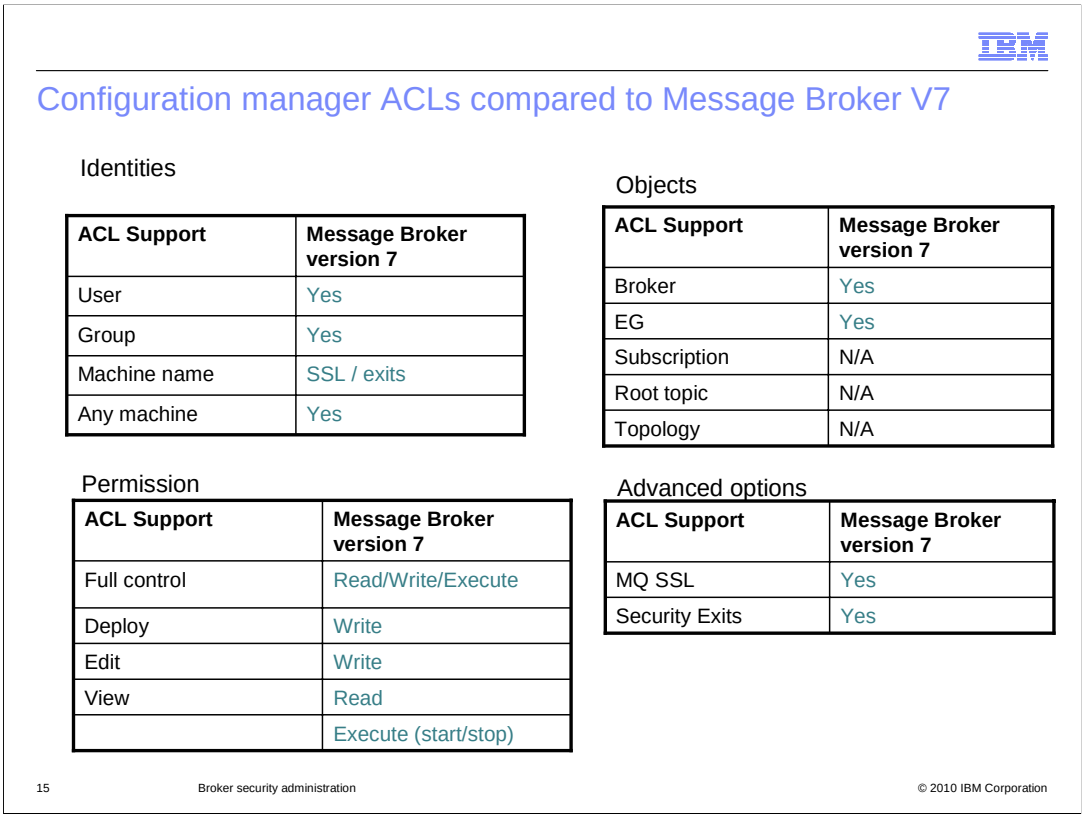
Command changes

- **mqsicreatebroker**
  - A new optional flag (-s) to specify that when a broker is created, security is enabled
  - If not specified, security will be off
- **mqsichangebroker**
  - A new optional flag (-s) to enable security on existing brokers
- **mqsideletebroker**
  - A new optional flag (-s) to perform the deletion of all SYSTEM.BROKER.AUTH.* queues
    - Default is to leave the queues there

This slide summarizes the changes in the commands that are related to security authorization.

The “-s” option is used to control security settings. This is used on the create broker and the change broker commands to enable or activate security.

On the delete broker command, it is used to control whether the authorization queues are deleted when the broker is deleted.



This slide summarizes the differences between the access control lists that were available in earlier versions of Message Broker, using the Configuration Manager to manage security, and the implementation in version 7.

For example, in the “Identities” tables, the ACL support enables specific authorities to be granted to a user or to a group, and this support is available with the new tools in version 7.

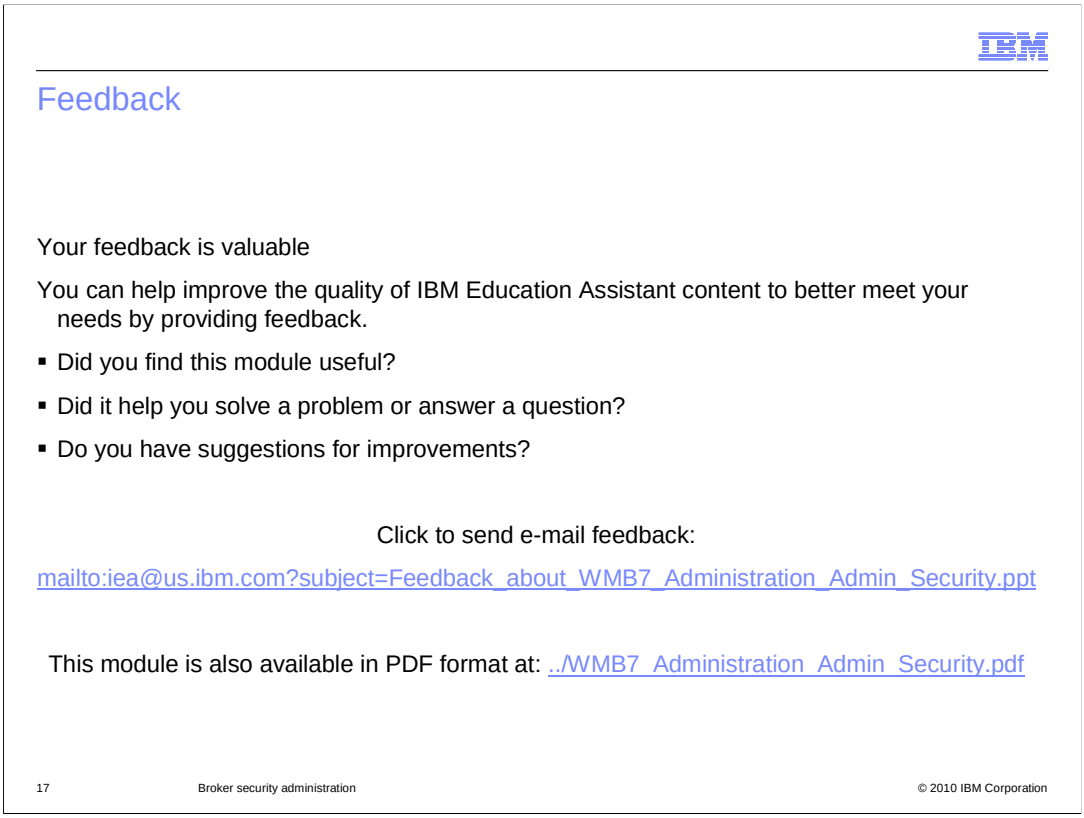
In the permissions table, you were able to grant different authorities such as full control, deploy or view. These map onto the equivalent functions in version 7, as shown in the table.

Summary

- Message Broker uses MQ permissions for security administration
- Message Broker B has read, write and execute authorities
- SYSTEM.BROKER.AUTH queues are used to check MQ permissions
- “mqsi” commands to enable and disable security
- MQ permissions changed without a broker restart

In summary, Message Broker version 7 has changed the way that security administration is implemented. This is because the configuration manager has been removed in version 7.

Security is now done using MQ queues, and MQ tools are used to manage the permissions. These permissions are set dynamically, and it is not necessary to restart any broker component for these to be activated.



You can help improve the quality of IBM Education Assistant content by providing feedback.

Trademarks, disclaimer, and copyright information