Unit 1. IBM MQ review

Estimated time

00:30

Overview

This unit reviews IBM MQ basic concepts and components. It also reviews the runtime options for IBM MQ on-premises and in the Cloud.

How you will check your progress

- Review questions

References

IBM Knowledge Center for IBM MQ V9

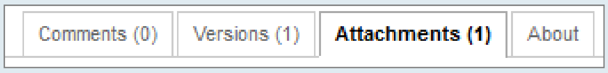
IBM Training

How to check online for course material updates

Note: If your classroom does not have internet access, ask your instructor for more information.

Instructions

1. Enter this URL in your browser: **ibm.biz**/CloudEduCourses
2. Find the product category for your course, and click the link to view all products and courses.
3. Find your course in the course list and then click the link.
4. The wiki page displays information for the course. If there is a course corrections document, this page is where it is found.
5. If you want to download an attachment, such as a course corrections document, click the **Attachments** tab at the bottom of the page.

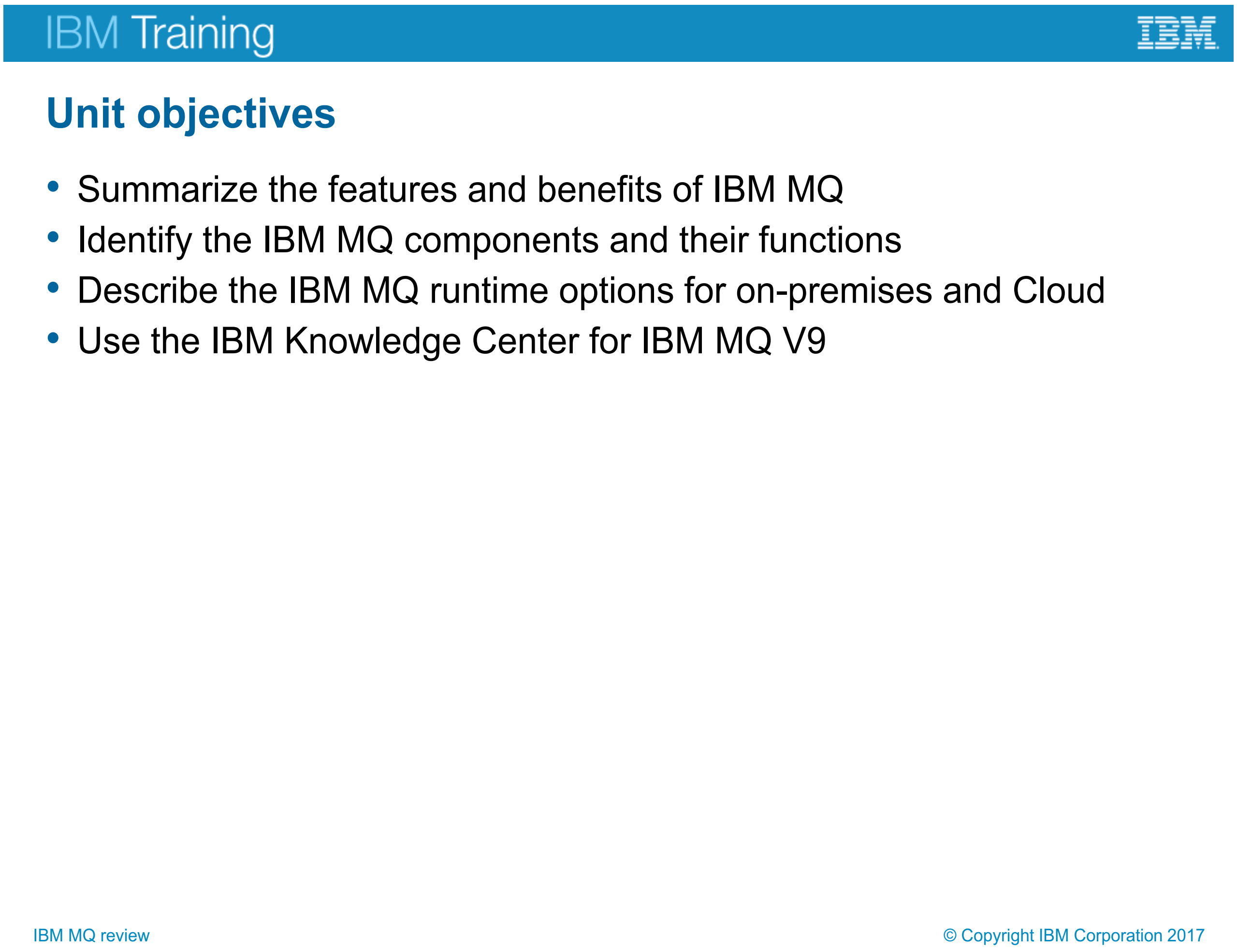


6. To save the file to your computer, click the document link and follow the prompts.

IBM MQ review

© Copyright IBM Corporation 2017

Figure 1-1. How to check online for course material updates



IBM Training

Unit objectives

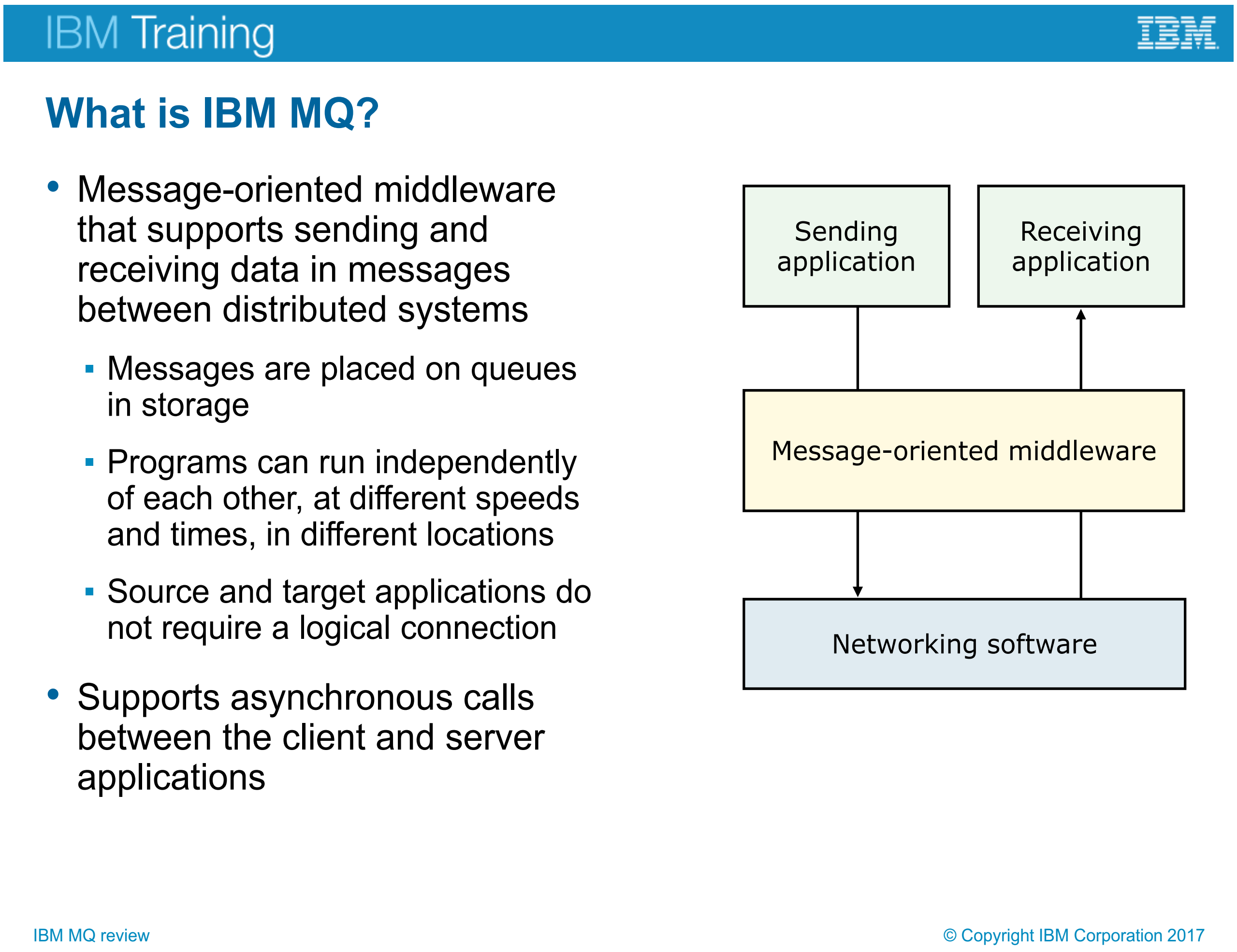
- Summarize the features and benefits of IBM MQ
- Identify the IBM MQ components and their functions
- Describe the IBM MQ runtime options for on-premises and Cloud
- Use the IBM Knowledge Center for IBM MQ V9

IBM MQ review

© Copyright IBM Corporation 2017

Figure 1-2. Unit objectives

Course WM103 or ZM103, *A Technical Introduction to IBM MQ V9*, is a prerequisite for this course; this unit is a review of previously acquired knowledge.



IBM Training

What is IBM MQ?

- Message-oriented middleware that supports sending and receiving data in messages between distributed systems
  - Messages are placed on queues in storage
  - Programs can run independently of each other, at different speeds and times, in different locations
  - Source and target applications do not require a logical connection
- Supports asynchronous calls between the client and server applications

IBM MQ review

© Copyright IBM Corporation 2017

Figure 1-3. What is IBM MQ?

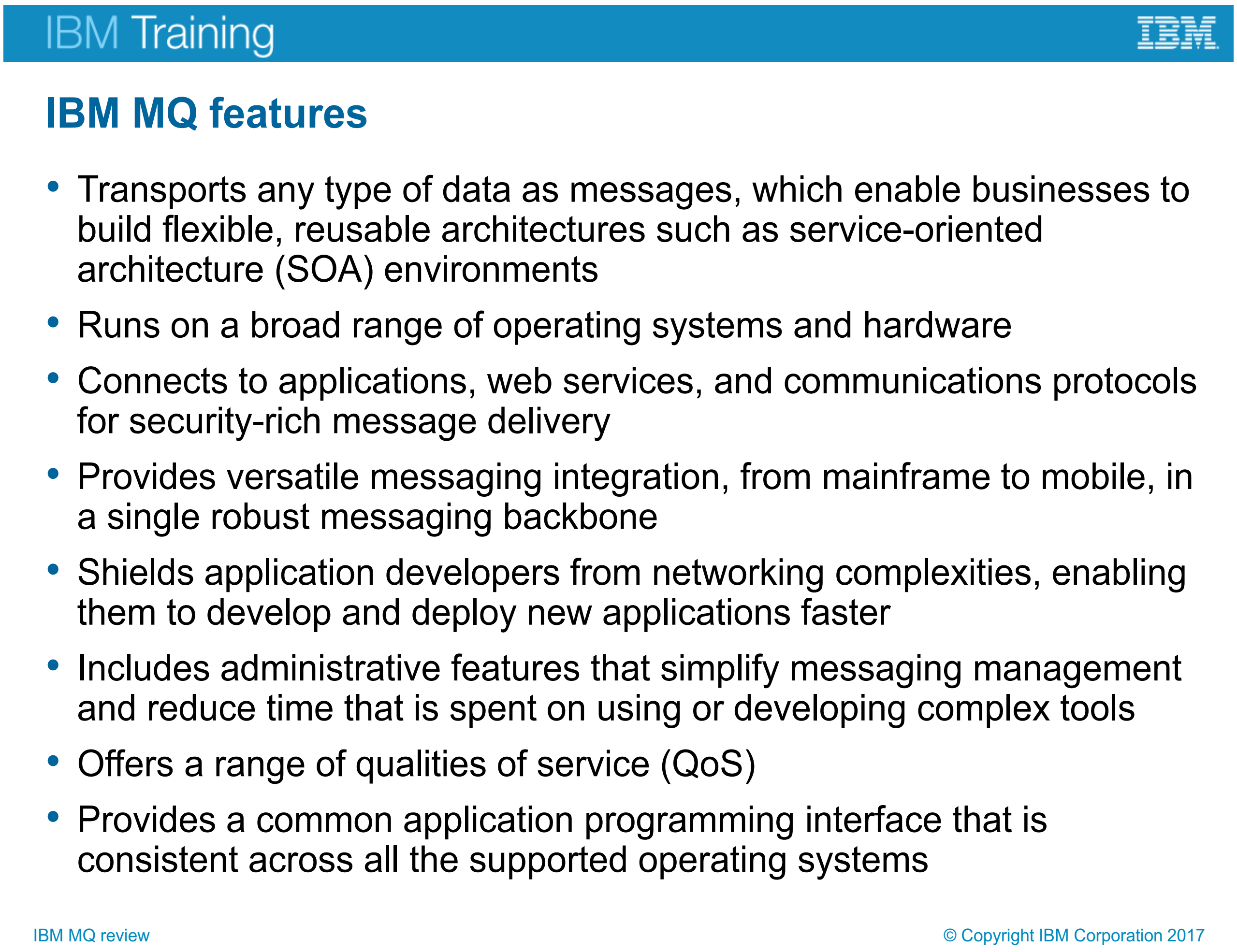
A key impediment to a flexible enterprise is traditional application connectivity, which combines business applications with one of many network application programming interfaces (API). This architecture requires knowledge of the networking conditions to adequately handle any communication problems.

With messaging-oriented middleware, such as IBM MQ, all the work of connecting, polling, handling failure, and implementing change is removed from the application. Instead, message-oriented middleware provides a dedicated middleware layer for handling connectivity. It is decoupled from the sending and receiving application, which provides the flexibility to react to business conditions.

IBM MQ creates a distributed communications layer that insulates the application developer from the details of the various operating systems and network interfaces. So, application modules can be distributed over heterogeneous operating systems.

In IBM MQ, data is packaged in *messages*, which are stored on *queues*. Queues allow programs to put and get messages as required so that programs can run independently of each other, at different speeds and times, and in different locations.

IBM MQ supports asynchronous calls between the client and server applications.



IBM Training

IBM MQ features

- Transports any type of data as messages, which enable businesses to build flexible, reusable architectures such as service-oriented architecture (SOA) environments
- Runs on a broad range of operating systems and hardware
- Connects to applications, web services, and communications protocols for security-rich message delivery
- Provides versatile messaging integration, from mainframe to mobile, in a single robust messaging backbone
- Shields application developers from networking complexities, enabling them to develop and deploy new applications faster
- Includes administrative features that simplify messaging management and reduce time that is spent on using or developing complex tools
- Offers a range of qualities of service (QoS)
- Provides a common application programming interface that is consistent across all the supported operating systems

IBM MQ review

© Copyright IBM Corporation 2017

Figure 1-4. IBM MQ features

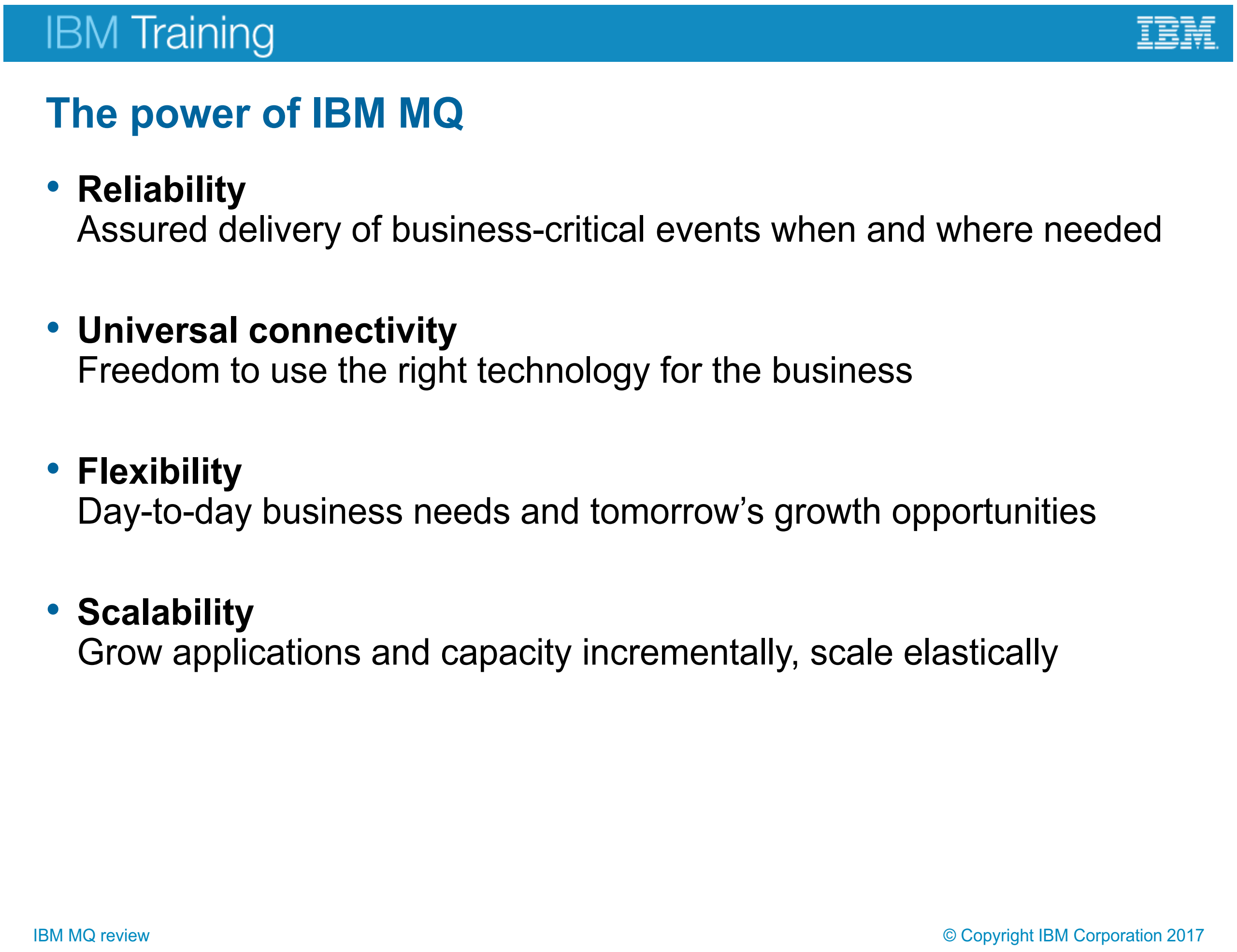
What is IBM MQ?

IBM MQ is enterprise messaging software that is used in the market for over 25 years.

IBM MQ provides asynchronous messaging for a wide range of systems. Applications that are indecipherable because they were written in an obsolete language can communicate with those applications that were written in the popular language last week.

IBM MQ communication is fast, and because the messaging is asynchronous, the business has the assurance that data is not lost. With IBM MQ, data arrives on time, in order, and once only.

IBM MQ includes administrative features and tools that simplify the management of the MQ components and the messages that it processes. It also offers a range of qualities of service (QoS).



IBM Training

The power of IBM MQ

- **Reliability**
  Assured delivery of business-critical events when and where needed
- **Universal connectivity**
  Freedom to use the right technology for the business
- **Flexibility**
  Day-to-day business needs and tomorrow's growth opportunities
- **Scalability**
  Grow applications and capacity incrementally, scale elastically

IBM MQ review

© Copyright IBM Corporation 2017

Figure 1-5. The power of IBM MQ

Effective messaging provides reliability, universal connectivity, flexibility, and scalability.

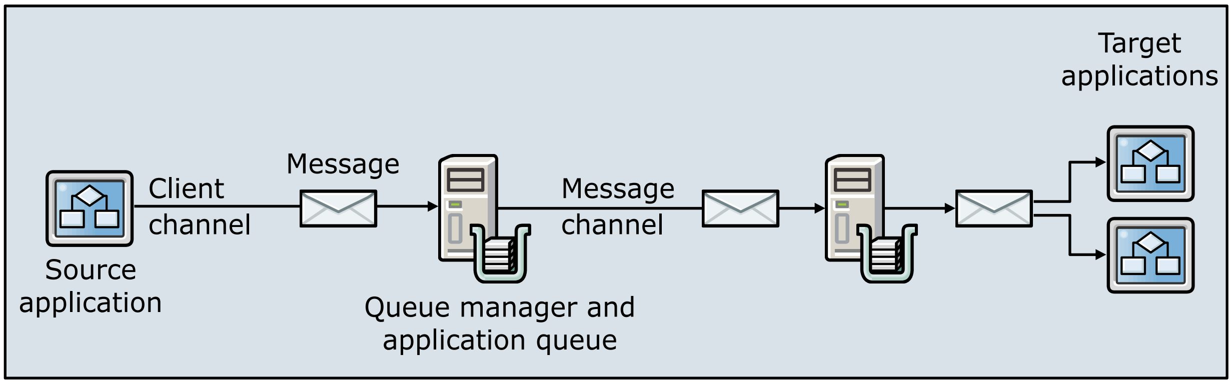
Effective messaging solves the following common business problems:

- If one application fails, many other applications fail. It costs the business a lot in downtime and recovery.
- An organization's applications run on different hardware and operating systems, and are written in different programming languages.
- A process was originally designed for one purpose, but now it must change to meet new requirements without breaking other applications.
- The business expands, and the capacity of the system can no longer cope with the workload demand.

IBM Training

IBM MQ components

- *Messages* that contain application data
- *Queues* that hold messages
- *Channels* that connect queue managers and client applications
- *Queue manager* that manages messages, queues, channels, and other IBM MQ services and resources



IBM MQ review

Figure 1-6. IBM MQ components

The main components of an IBM MQ configuration are queue managers, messages, queues, and channels. Each of these components is described in more detail in this unit.

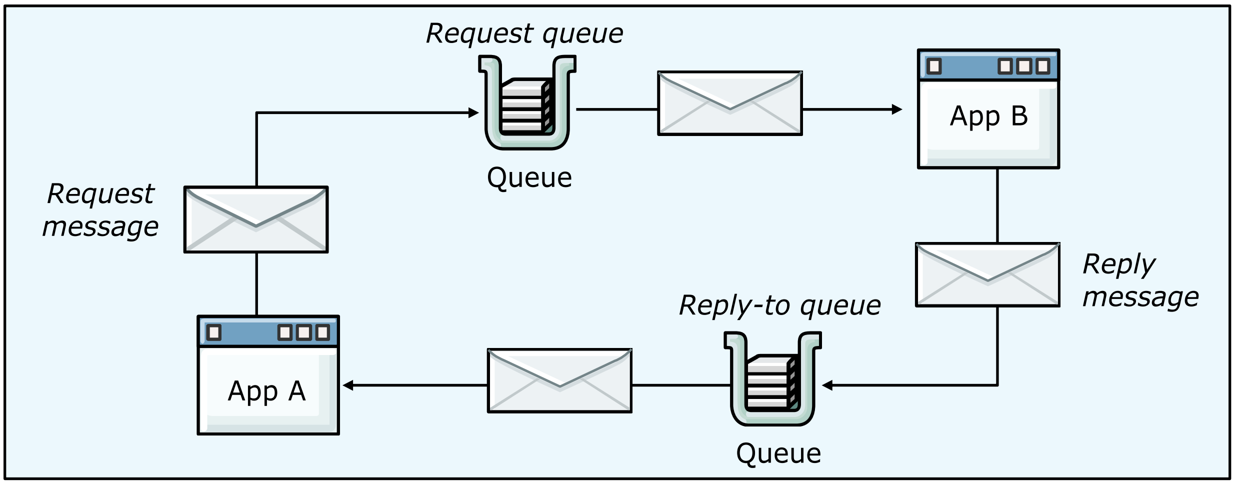
An *MQ server* controls the computer that contains the IBM MQ configuration. Computers that contain client applications that must connect to the MQ server must also have an *MQ client* installation. It is possible to have both an MQ server and an MQ client installation on the same computer.

The difference between an MQ server and an MQ client is discussed at the end of this unit. Unless the term MQ client is specified, the content in this unit applies to MQ server installations.

IBM Training

IBM MQ functional overview

- IBM MQ uses messages and queues to provide program-to-program communication
- IBM MQ provides:
  - Assured message delivery
  - Time-independent processing
  - Application parallelism



IBM MQ review © Copyright IBM Corporation 2017

Figure 1-7. IBM MQ functional overview

IBM MQ uses messages and queues to provide program-to-program communication. The communicating applications can be on the same system, or distributed across a network of IBM and non-IBM systems.

In general, multiple applications can put messages on the same queue to request the same service. The application that serves the queue gets each message and responds to it.

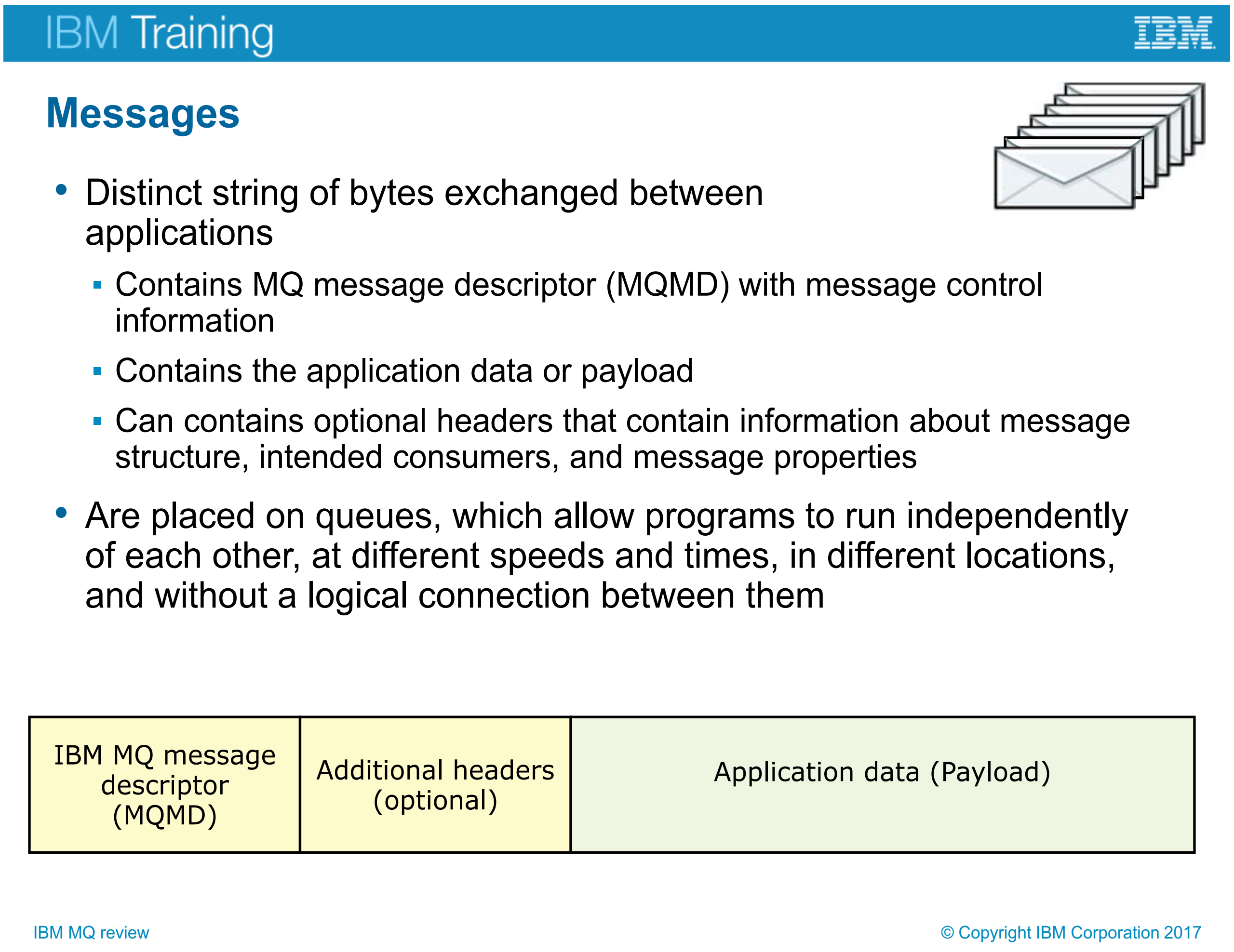
The example in the figure shows a simple request-reply scenario that uses messages and queues. In this example, a client application (App A) puts a message onto a request queue. Another application (App B) gets the message from the request queue and processes the request. The reply message is put on the reply-to queue. The original requester (App A) can then get the reply message from the reply-to queue.

In this example, the original request contains a message descriptor that provides the name of the reply-to queue for the reply message. Each requesting application can have its own reply-to queue so that each client application can receive its replies separately from other client applications.

The message descriptor also contains a message identifier that the requesting application can use to correlate a reply with a request.

IBM MQ can transfer data with assured delivery; messages do not get lost, even when a system failure occurs. MQ does not duplicate message delivery.

With IBM MQ, communicating applications do not have to be active at the same time. An application is divided into discrete functional modules that communicate with each other by using messages. In this way, the modules can run on different systems, be scheduled at different times, or they can act in parallel.



IBM Training

Messages

- Distinct string of bytes exchanged between applications
  - Contains MQ message descriptor (MQMD) with message control information
  - Contains the application data or payload
  - Can contains optional headers that contain information about message structure, intended consumers, and message properties
- Are placed on queues, which allow programs to run independently of each other, at different speeds and times, in different locations, and without a logical connection between them

IBM MQ message descriptor (MQMD)	Additional headers (optional)	Application data (Payload)

IBM MQ review

© Copyright IBM Corporation 2017

Figure 1-8. Messages

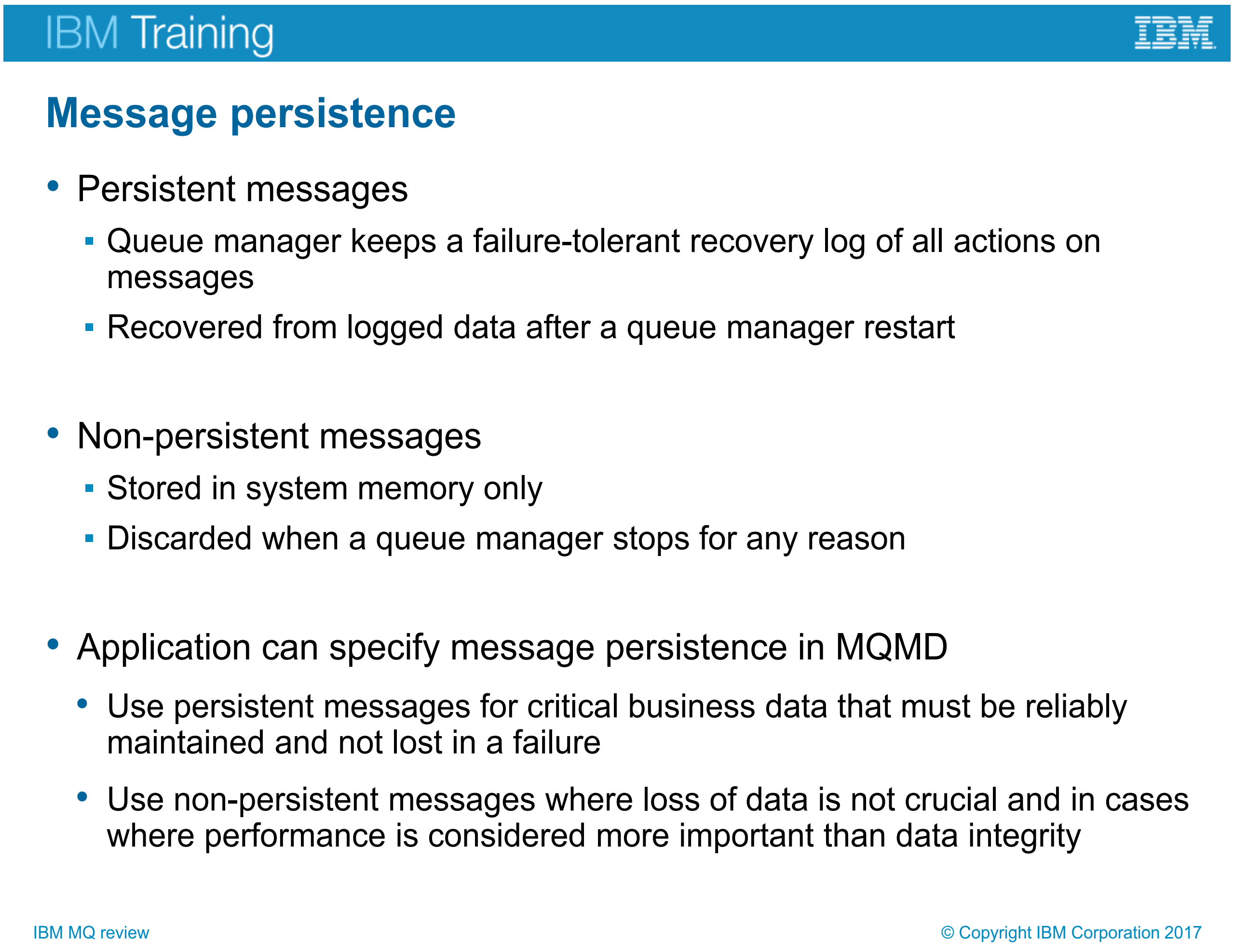
The application data that MQ exchanges is sent in the form of messages. A *message* is any information that one application uses to communicate to another. A message can convey a request for a service, or it can be a reply to a request. A message might also report on the progress of another message: to confirm its arrival or report on an error, for example. A message can also carry information for which no reply is expected.

Messages consist of the MQ message descriptor (MQMD) and the payload. Optionally the application can also provide message properties.

The MQMD contains information about the message. Objects are identified by the MQMD when addressed from a program. The MQMD contains information about the sending application, for example.

The message data portion of an MQ message contains the actual payload, such as a bank transaction or healthcare record.

The sending application supplies both the message descriptor and the application data when it puts a message on a queue. The application that puts the messages on the queue sets some of the fields in the message descriptor; the queue manager sets others on behalf of the application. Both the message descriptor and the application data are returned to the application that gets the message from the queue.



IBM Training

Message persistence

- Persistent messages
  - Queue manager keeps a failure-tolerant recovery log of all actions on messages
  - Recovered from logged data after a queue manager restart
- Non-persistent messages
  - Stored in system memory only
  - Discarded when a queue manager stops for any reason
- Application can specify message persistence in MQMD
  - Use persistent messages for critical business data that must be reliably maintained and not lost in a failure
  - Use non-persistent messages where loss of data is not crucial and in cases where performance is considered more important than data integrity

IBM MQ review © Copyright IBM Corporation 2017

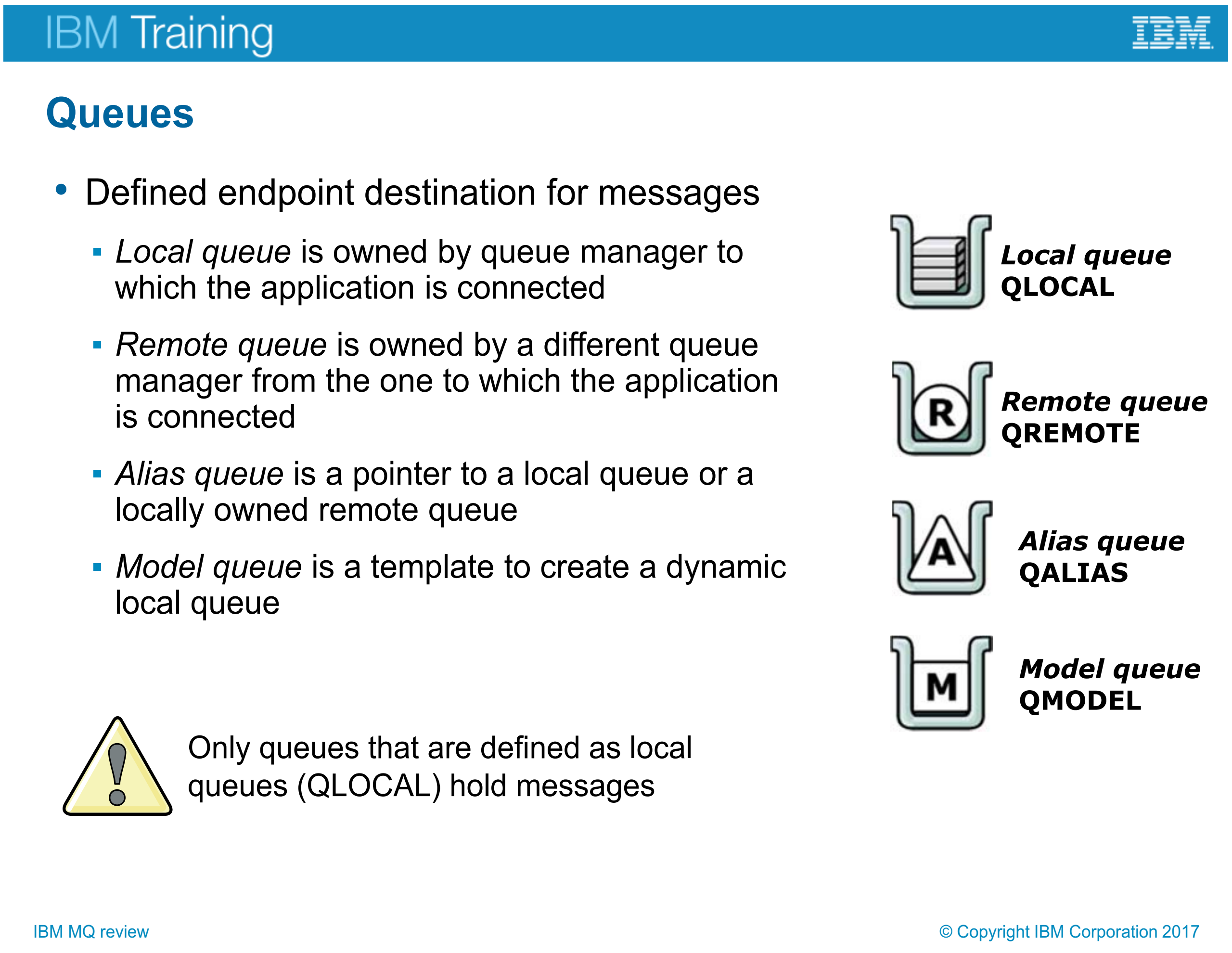
Figure 1-9. Message persistence

The MQMD contains a persistence attribute that controls whether a message survives restarts of the queue manager.

If the application identifies the message as persistent, the queue manager keeps a failure-tolerant recovery log of all actions on that message. If a queue manager restarts before the message is delivered to its target, the message is recovered from the logged data. If an application contains critical business data that must be reliably maintained, it should identify a message as persistent. Examples of persistent messages are stock trades or banking transactions.

If the application identifies the message as nonpersistent, the message is stored in system memory only. If a queue manager restarts before the message is delivered to its target, the message is discarded. An application can identify a message as nonpersistent when the loss of data is not crucial.

Persistence is a critical attribute that controls whether a message survives restarts of the queue manager. What if it carries time-sensitive information? What if it contains an important time-independent notification? Before you rush to make a message persistent, it is important that you understand the use for this data.



IBM Training

Queues

- Defined endpoint destination for messages
  - *Local queue* is owned by queue manager to which the application is connected
  - *Remote queue* is owned by a different queue manager from the one to which the application is connected
  - *Alias queue* is a pointer to a local queue or a locally owned remote queue
  - *Model queue* is a template to create a dynamic local queue

Only queues that are defined as local queues (QLOCAL) hold messages

Local queue
QLOCAL

Remote queue
QREMOTE

Alias queue
QALIAS

Model queue
QMODEL

IBM MQ review

© Copyright IBM Corporation 2017

Figure 1-10. Queues

An MQ queue is a named object on which applications can put messages, and from which applications can get messages. Messages are stored on a queue, so that if the putting application is expecting a reply to its message, it can do other work while it waits for the reply. Applications access a queue by using the Message Queue Interface (MQI).

MQ supports different types of queues for storing, identifying, and moving messages in the MQ network. This figure summarizes the types of queues that the MQ supports.

Messages are stored on local queues, which are owned by the queue manager to which the application connects.

Remote queues and alias queues are pointers to other queues but do not hold messages. Messages can be directed to remote or alias queues, but the final target is a local queue.

It is not possible to retrieve messages from a remote queue or an alias queue; messages are always in the target local queue that is identified in the definition.

A model queue is a template for a dynamic local queue.

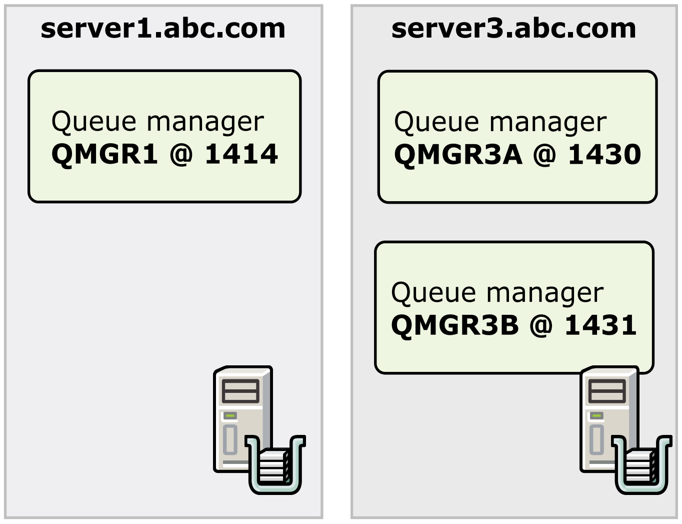
A queue is owned by a queue manager, and that queue manager can own many queues. However, each queue must have a name that is unique within that queue manager.

The graphics on this figure are used throughout this course to identify the different types of queues.

IBM Training

Queue manager

- Provides services for accepting and delivering messages
- Maintains queues of all messages that are waiting to be processed or routed
- System program that owns the resources that it services
- Requires a name and TCP/IP listener port
- A server can host more than one queue manager
  - Queue managers that share a server require different names and listener ports


server1.abc.com
Queue manager QMGR1 @ 1414
server3.abc.com
Queue manager QMGR3A @ 1430
Queue manager QMGR3B @ 1431

IBM MQ review

Figure 1-11. Queue manager

Queue managers are the main components in an MQ messaging network. Queue managers host the other objects in the network, such as the queues and the channels that connect the queue managers together.

Queue managers define the properties of MQ objects. The values of these properties affect how MQ processes these objects. You create and manage objects by using MQ commands and interfaces.

A server can contain more than one queue manager but each queue manager on the server must have a unique name.

The queue manager communicates by using a TCP connection. In a distributed environment, each queue manager is assigned a unique TCP/IP port. The default TCP port for a queue manager is 1414. The port number 1414 is assigned to MQ by the Internet Assigned Numbers Authority. When the server contains more than one queue manager, each queue manager must have a different port number.

In the example, the server that is named server1.abc.com contains one queue manager that is named QMGR1, which is listening on TCP port 1414. The server that is named server3.abc.com contains two queue managers. The queue manager that is named QMGR3A is listening on TCP port 1430. The queue manager that is named QMGR3B is listening on TCP port 1431.

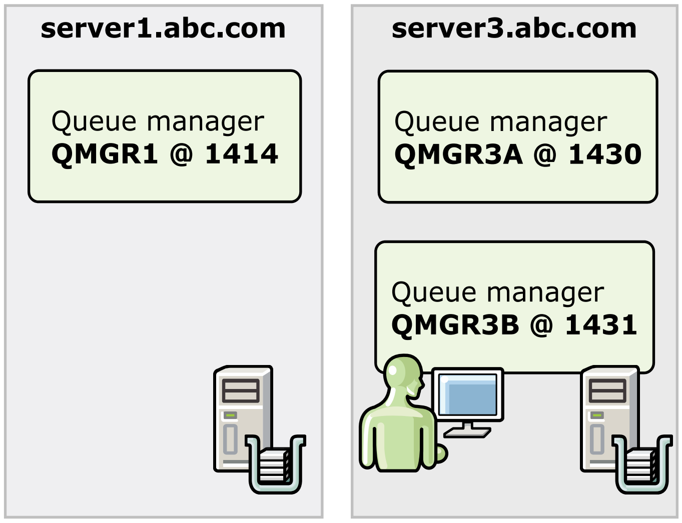
IBM Training

Queue manager terminology

- *Local queue manager* is the currently referenced queue manager
- All other queue managers are *remote queue managers*

Example:

- If working on QMGR3B, then QMGR3B is the local queue manager, and QMGR1 and QMGR3A are remote queue managers
- Objects and resources that are defined in QMGR3B are said to be "locally owned"



IBM MQ review

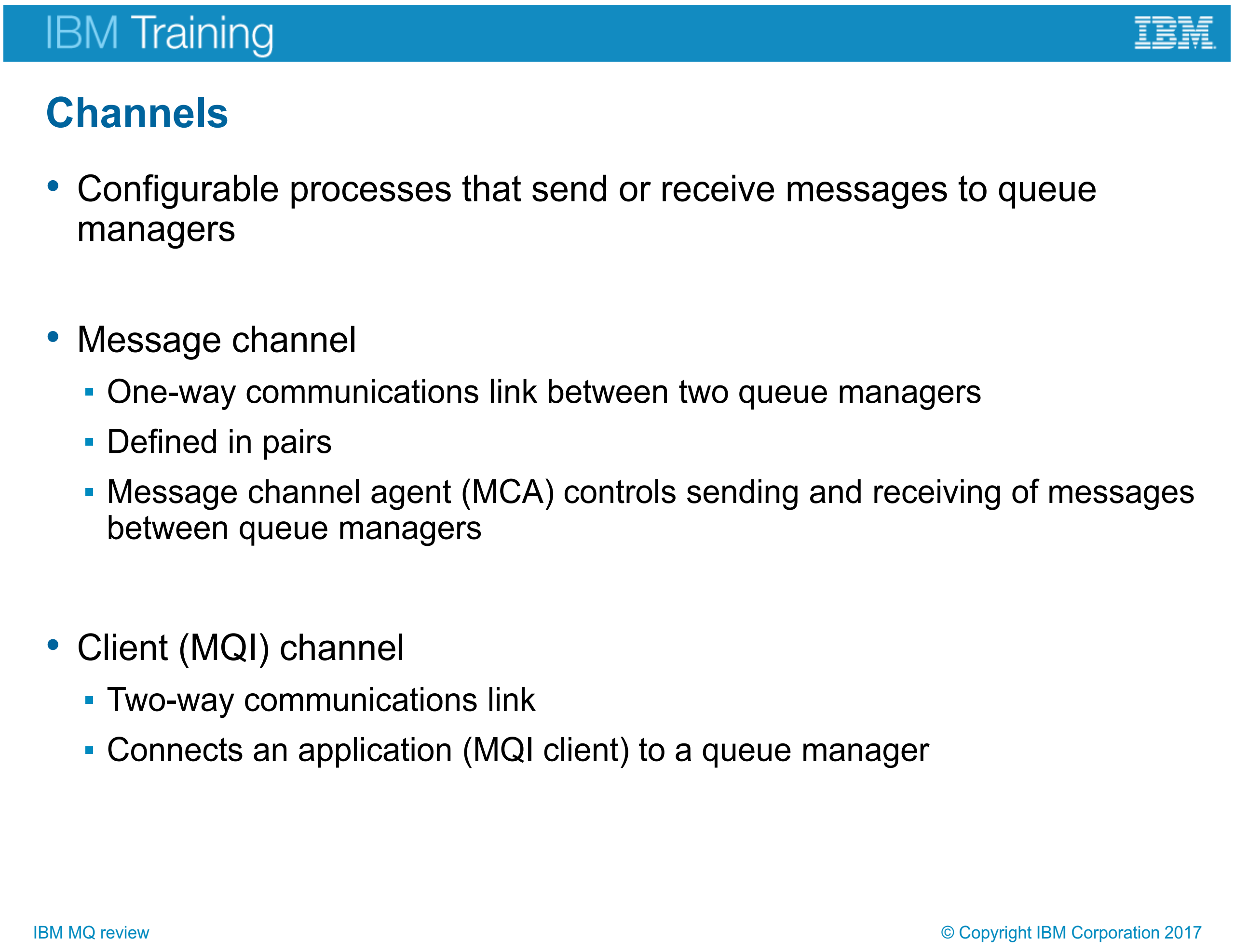
© Copyright IBM Corporation 2017

Figure 1-12. Queue manager terminology

To avoid confusion, you must understand what is meant by *remote* when you reference a queue manager in the network.

Whether a queue manager is local or remote depends on whether the administrator or application is directly connected to a queue manager. A local queue manager is the currently referenced queue manager; a remote queue manager is any queue manager that is not currently referenced.

The local queue manager designation changes when work moves to another queue manager. For example, if an administrator is connected to queue manager QMGR3B, then that queue manager is the local queue manager and the other queue managers (QMGR3A and QMGR1) are remote queue managers. If the administrator is connected to the queue manager that is named QMGR3A, the other queue managers (QMGR3B and QMGR1) are the remote queue managers.



IBM Training

Channels

- Configurable processes that send or receive messages to queue managers
- Message channel
  - One-way communications link between two queue managers
  - Defined in pairs
  - Message channel agent (MCA) controls sending and receiving of messages between queue managers
- Client (MQI) channel
  - Two-way communications link
  - Connects an application (MQI client) to a queue manager

IBM MQ review © Copyright IBM Corporation 2017

Figure 1-13. Channels

Channels are the processes that move messages to and from queue managers.

A *message channel* provides a one-way communications link between queue managers. Message channels are used in distributed queuing to move messages from one queue manager to another.

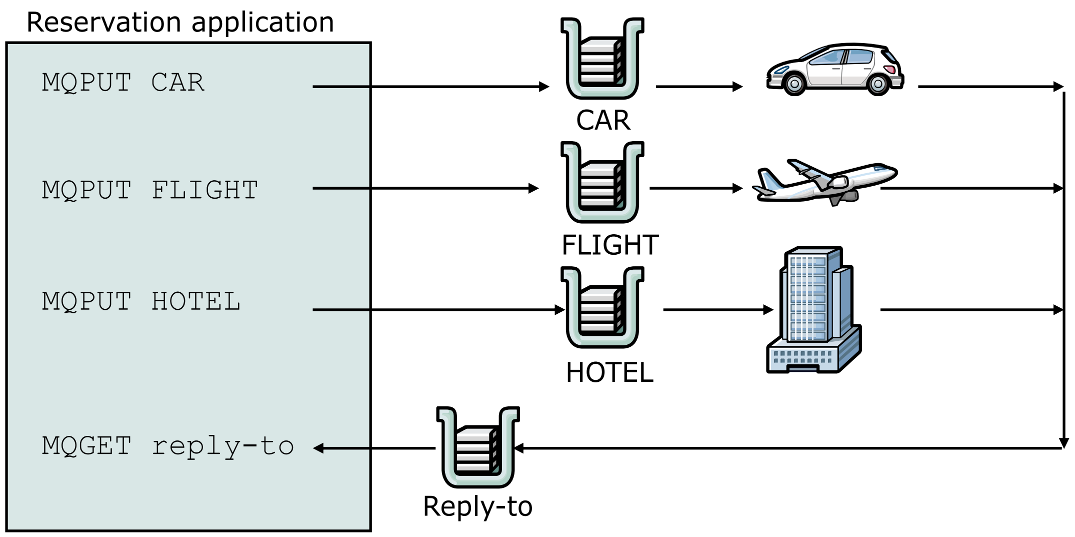
When you define message channels, you must define one channel object at the queue manager that is to send messages. You must also define another, complementary channel, at the queue manager that is to receive the messages. Sender-receiver channels are the most commonly used distributed message channels.

A message channel connects two queue managers by using message channel agents (MCAs).The MCA on one end takes messages from local transmission queue and puts them on communication link. The MCA on other end receives the messages and puts them on target queue.

A client channel, also referred to as an *MQI channel*, provides a two-way communications link between a client application and a queue manager.

IBM Training

Parallel processing



- Message descriptor provides name of reply-to queue
- Requests are not serialized
- Replies are consolidated
- Transactions have shorter elapsed time

IBM MQ review © Copyright IBM Corporation 2017

Figure 1-14. Parallel processing

In general, multiple applications can put messages on the same queue to request the same service. The application that serves the queue gets each message and responds to it.

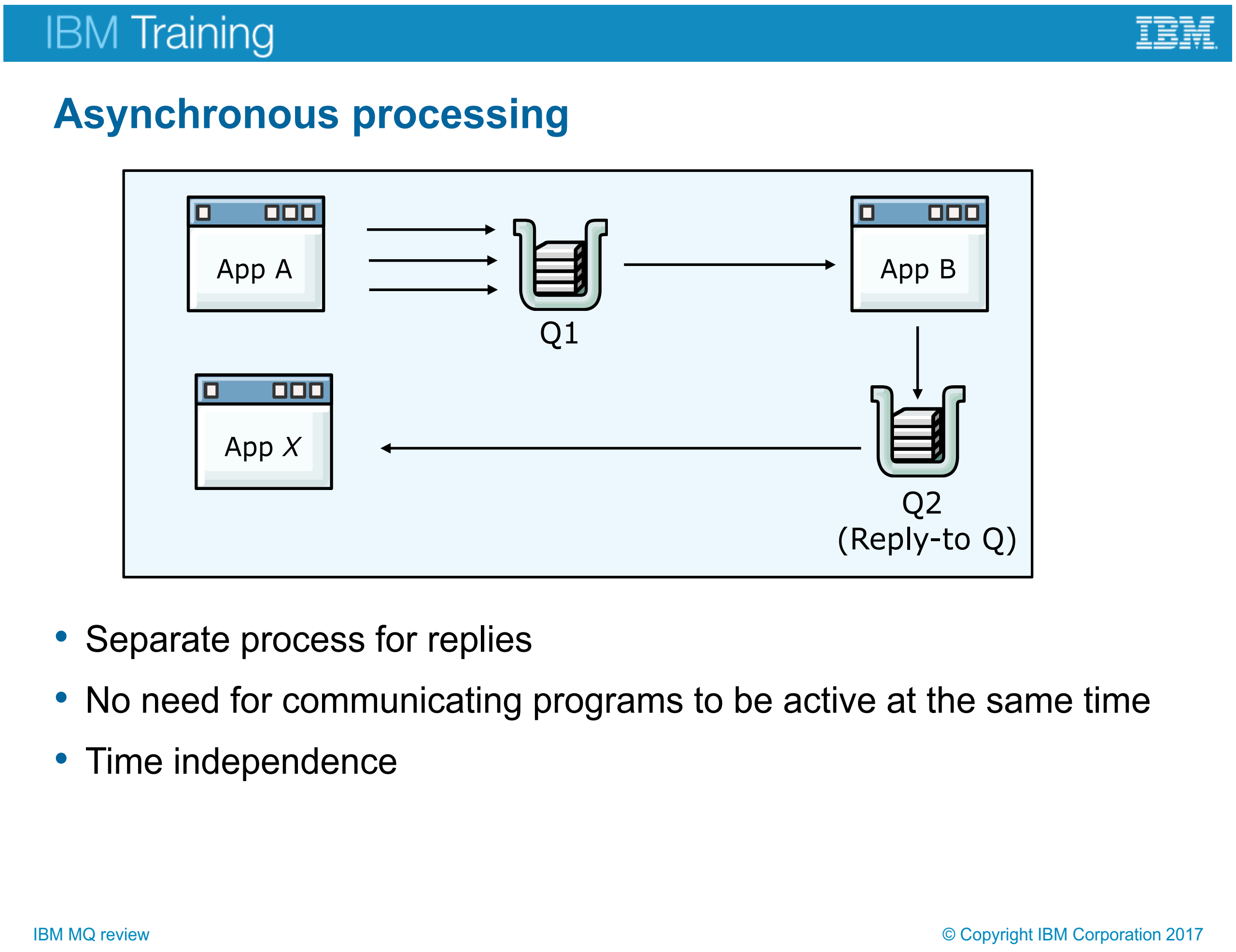
Parallel processing allows an application to send several requests without waiting for a reply to one request before sending the next. All the requests can then be processed in parallel.

The application can process the replies when they are all received, and produce a consolidated answer. The program logic might also specify what to do when only a partial set of replies is received within a specified period.

A more complex application might involve sending a number of requests to different servers. In the example in the figure, a travel agent uses an application to arrange a trip for a customer. The application must make a number of requests, such as making a rental car, airplane, or hotel reservation. By using queues, these steps do not have to be completed serially.

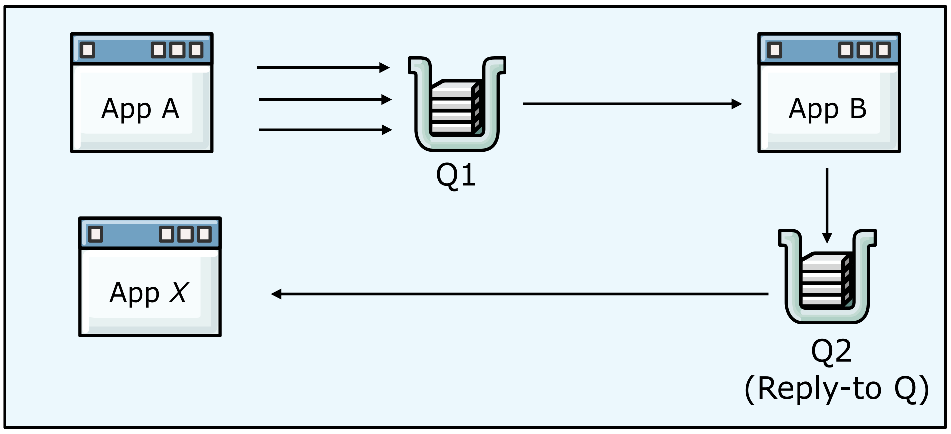
In this example, the message descriptor provides the name of the reply-to queue. The reply-to queue informs the server application where to put the reply message. In this way, the client application can receive its replies separately from other client applications.

The application can process the replies when they are all received, and produce a consolidated answer. The program logic might also specify what to do when only a partial set of replies is received within a specified period.



IBM Training

Asynchronous processing



- Separate process for replies
- No need for communicating programs to be active at the same time
- Time independence

IBM MQ review

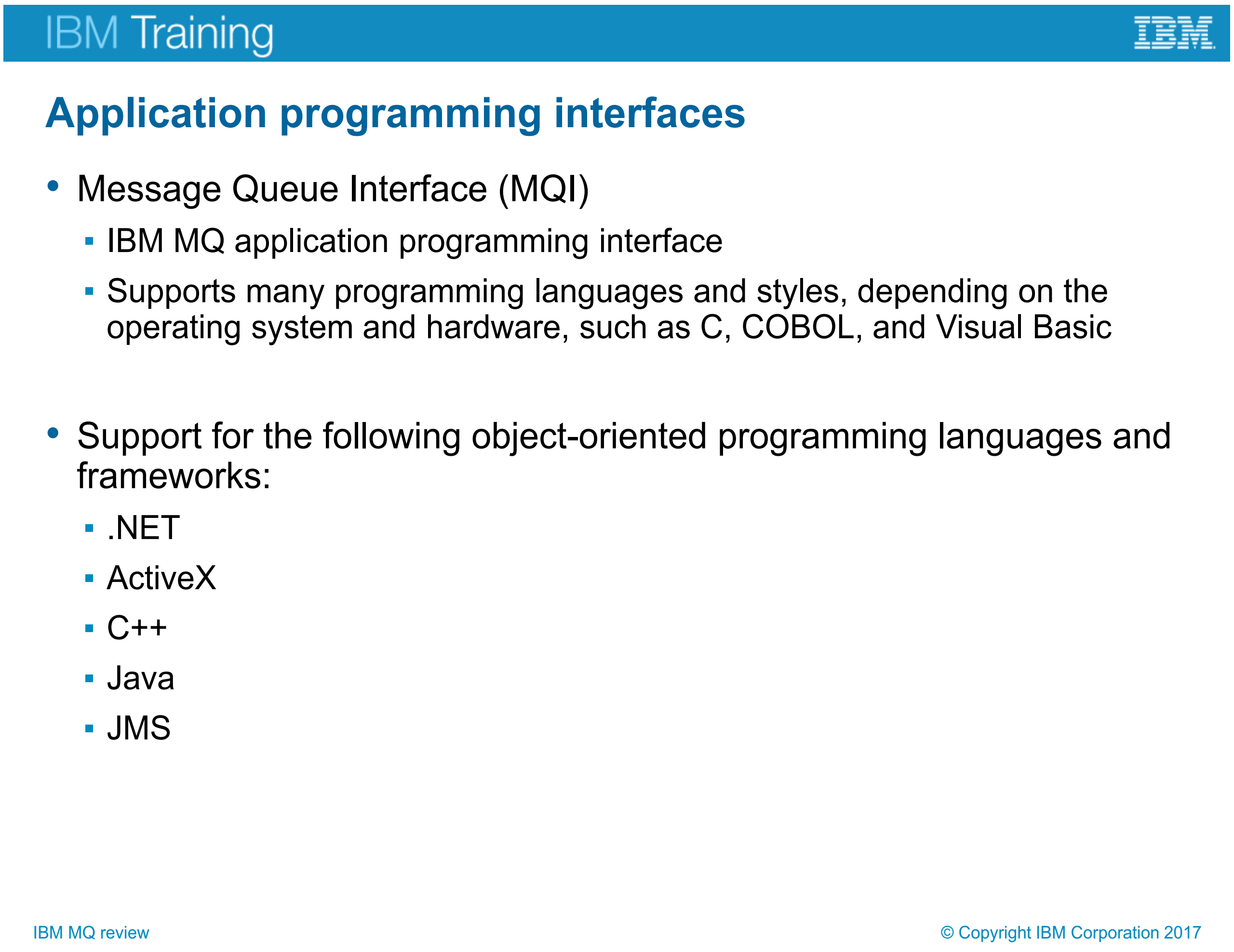
© Copyright IBM Corporation 2017

Figure 1-15. Asynchronous processing

In the asynchronous message processing model, instead of waiting for a reply to its first message, App A continues to send further requests to App B. In a separate process, App X receives the replies when they arrive on the reply-to queue (Q2).

In the asynchronous message processing model, App A is not dependent on App B to be running when the requests are sent. It can continue to do work even when App B is stopped.

In this model, it is expected that App X gets the reply messages but not necessarily at the same time that App B puts them on the reply-to queue. This scenario illustrates another of the major benefits of IBM MQ, which is *time independence*.



IBM Training

Application programming interfaces

- Message Queue Interface (MQI)
  - IBM MQ application programming interface
  - Supports many programming languages and styles, depending on the operating system and hardware, such as C, COBOL, and Visual Basic
- Support for the following object-oriented programming languages and frameworks:
  - .NET
  - ActiveX
  - C++
  - Java
  - JMS

IBM MQ review

Figure 1-16. Application programming interfaces

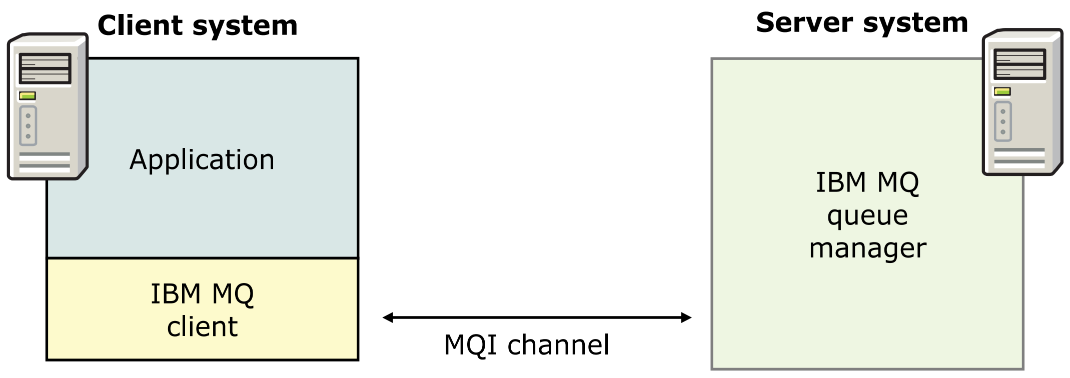
You can develop applications to send and receive messages, and to manage your queue managers and related resources. MQ supports applications that are written in procedural languages, and object-oriented languages and frameworks.

MQ supports an application programming interface. On distributed operating systems, MQ supports C, Visual Basic (on Windows), and COBOL. These procedural languages use the MQI to access MQ services.

MQ also supports the following object-oriented programming languages and frameworks: .NET, ActiveX, C++, Java, and JMS. These languages and frameworks use the IBM MQ Object Model. The IBM MQ Object Model provides classes that provide the same functions as the MQI calls and structures, but that are a more natural way of programming in an object-oriented environment. Some of the languages and frameworks that use the IBM MQ Object Model provide functions that are not available to the procedural languages that use the MQI.

IBM Training

IBM MQ client



- IBM MQ component on a system without a queue manager
  - Communicates with queue manager by an MQI channel
  - Allows an application that runs on the client system to connect to a queue manager that is running on another system

Figure 1-17. IBM MQ client

An MQ client is a component of MQ that allows an application that is running on one system to send MQI calls to a queue manager that is running on another system.

The *client connection* receives the input parameters of an MQI call from the application. It then sends the input parameters over a communications connection to the *server connection* on the same system as the queue manager. The server connection then sends the MQI call to the queue manager on behalf of the application. After the queue manager completes the MQI call, the server connection sends the output parameters of the callback to the client connection, which then passes them onto the application.

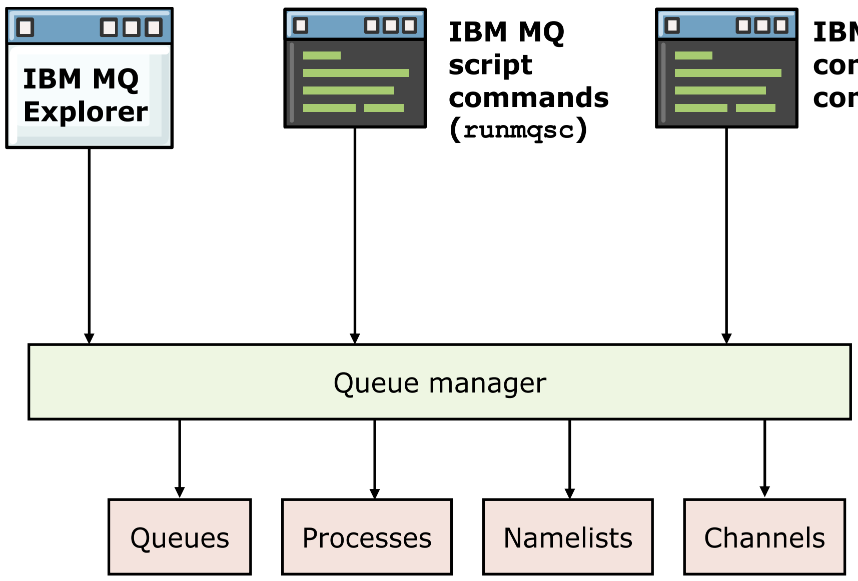
The combination of a client connection, a server connection, and a communications connection between them (for example, a TCP connection) is called an *MQI channel*.

The full range of MQI calls and options are available to a client application.

You learn more about MQ clients in Unit 7, “IBM MQ clients”.

IBM Training

IBM MQ administration interfaces



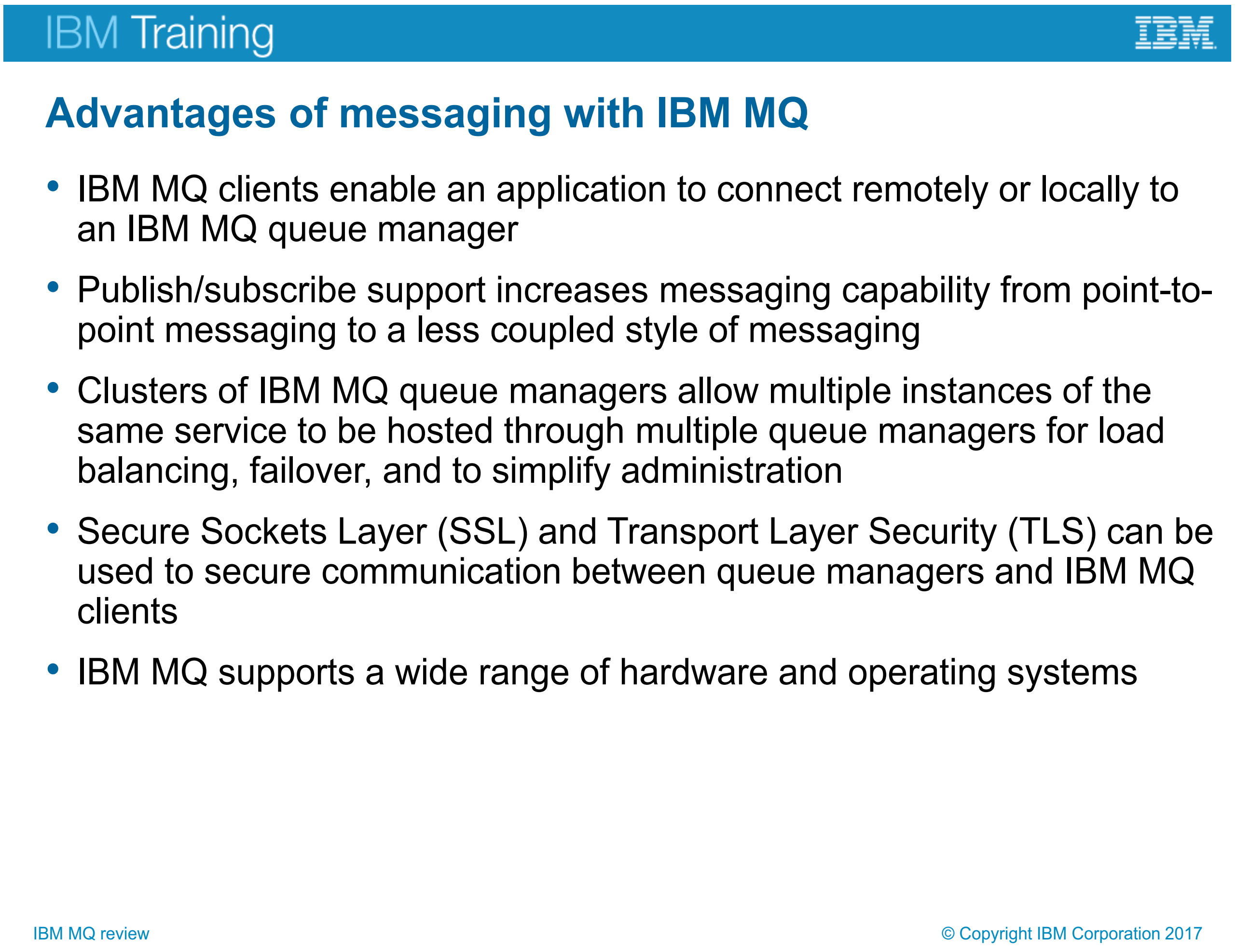
IBM MQ review © Copyright IBM Corporation 2017

Figure 1-18. IBM MQ administration interfaces

IBM MQ provides three primary interfaces for managing MQ queue managers and queue manager resources such as queues, processes, namelists, and channels.

IBM MQ Explorer is a graphical user interface that runs on Linux and Windows. It can connect to, control, and configure MQ on all operating systems. MQ Explorer is described in detail later in this course.

MQ script commands (MQSC) and MQ control commands are command-line interfaces that used to create and modify queue managers and queue manager resources. MQSC and MQ control commands are described in more detail throughout this course.



IBM Training

Advantages of messaging with IBM MQ

- IBM MQ clients enable an application to connect remotely or locally to an IBM MQ queue manager
- Publish/subscribe support increases messaging capability from point-to-point messaging to a less coupled style of messaging
- Clusters of IBM MQ queue managers allow multiple instances of the same service to be hosted through multiple queue managers for load balancing, failover, and to simplify administration
- Secure Sockets Layer (SSL) and Transport Layer Security (TLS) can be used to secure communication between queue managers and IBM MQ clients
- IBM MQ supports a wide range of hardware and operating systems

IBM MQ review © Copyright IBM Corporation 2017

Figure 1-19. Advantages of messaging with IBM MQ

This figure lists some of the IBM MQ features that make communication and configuration easier.

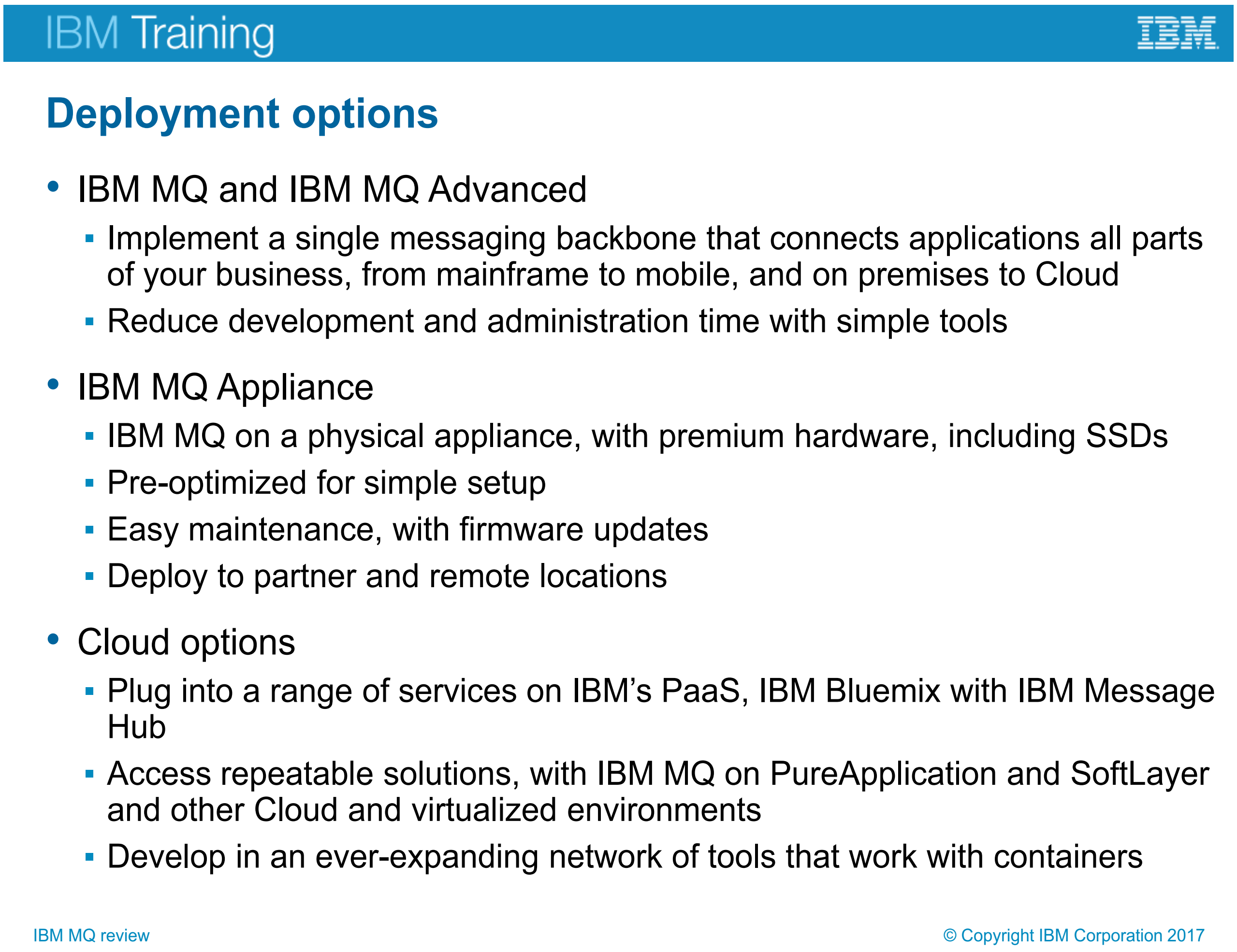
A separate MQ client component enables an application to connect remotely or locally to an MQ queue manager.

IBM MQ supports two primary communication methods: point-to-point and publish/subscribe. In publish/subscribe communication, a publishing application publishes a copy of each message and delivers it to every application that subscribes to the publication. With publish/subscribe messaging, you can decouple the provider of information from the consumers of that information. The sending application and receiving application do not need to know as much about each other for the information to be sent and received. Publish/subscribe is described in more detail in course WM213, *IBM MQ V9 Advanced System Administration on Distributed.*

An MQ cluster is a collection of queue managers that can be on different servers and operating systems, and typically serve a common application. Every queue manager in the cluster can make the queues that they host available to every other queue manager in the cluster, without the need for remote queue definitions. Queue manager clusters are described in more detail in Unit 12, “Introduction to queue manager clusters”.

MQ supports SSL and TLS for security channels between queue managers and between queue managers and MQ clients. Security is described in more detail in Unit 10, “Implementing basic security in IBM MQ”.

MQ runs on wide variety of operating systems and hardware. MQ installation options are described in more detail in Unit 2, “IBM MQ installation and deployment options”.



IBM Training

Deployment options

- IBM MQ and IBM MQ Advanced
  - Implement a single messaging backbone that connects applications all parts of your business, from mainframe to mobile, and on premises to Cloud
  - Reduce development and administration time with simple tools
- IBM MQ Appliance
  - IBM MQ on a physical appliance, with premium hardware, including SSDs
  - Pre-optimized for simple setup
  - Easy maintenance, with firmware updates
  - Deploy to partner and remote locations
- Cloud options
  - Plug into a range of services on IBM's PaaS, IBM Bluemix with IBM Message Hub
  - Access repeatable solutions, with IBM MQ on PureApplication and SoftLayer and other Cloud and virtualized environments
  - Develop in an ever-expanding network of tools that work with containers

IBM MQ review © Copyright IBM Corporation 2017

Figure 1-20. Deployment options

This figure summarizes the development options for IBM MQ.

IBM MQ is messaging middleware that simplifies and accelerates the integration of diverse applications and business data across multiple platforms. It uses message queues to facilitate the exchange of information between applications, systems, services, and files and simplifies the creation and maintenance of business applications.

IBM MQ Advanced is IBM MQ plus managed file transfer, advanced message security, and telemetry. It provides lightweight messaging between the mainframes and mobile, and machine-to-machine connectivity, helping to reduce business costs and complexities, while it speeds time to value.

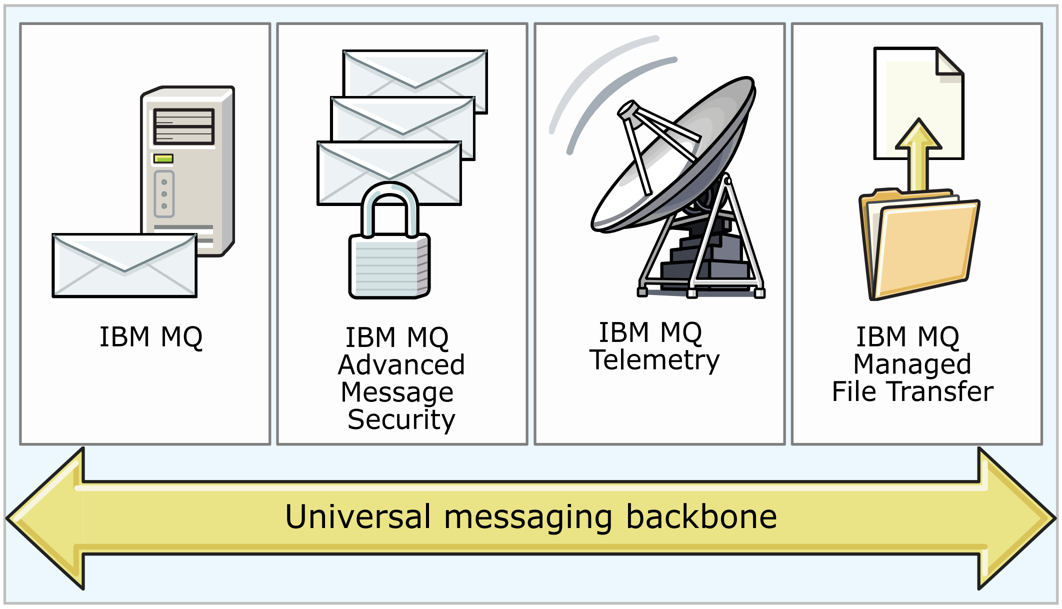
IBM MQ and IBM MQ Advanced can be installed on a server network in your enterprise.

The IBM MQ Appliance provides an optimized version of MQ that runs in a hardware appliance. It offers MQ capability in a hardware appliance and provides more features and some appliance benefits.

You can extend IBM MQ to the Cloud by using IBM's PaaS (platform as a service) such as IBM PureApplication System, IBM Bluemix, IBM SoftLayer, and other Cloud and virtualized environments.

IBM Training

IBM MQ Advanced


IBM MQ
IBM MQ Advanced Message Security
IBM MQ Telemetry
IBM MQ Managed File Transfer
Universal messaging backbone

- Strengths of IBM MQ
- End-to-end encryption of data with IBM MQ Advanced Message Security
- Transfer of files as messages with IBM MQ Message File Transfer
- Real-time connection to sensors and mobile with IBM MQ Telemetry

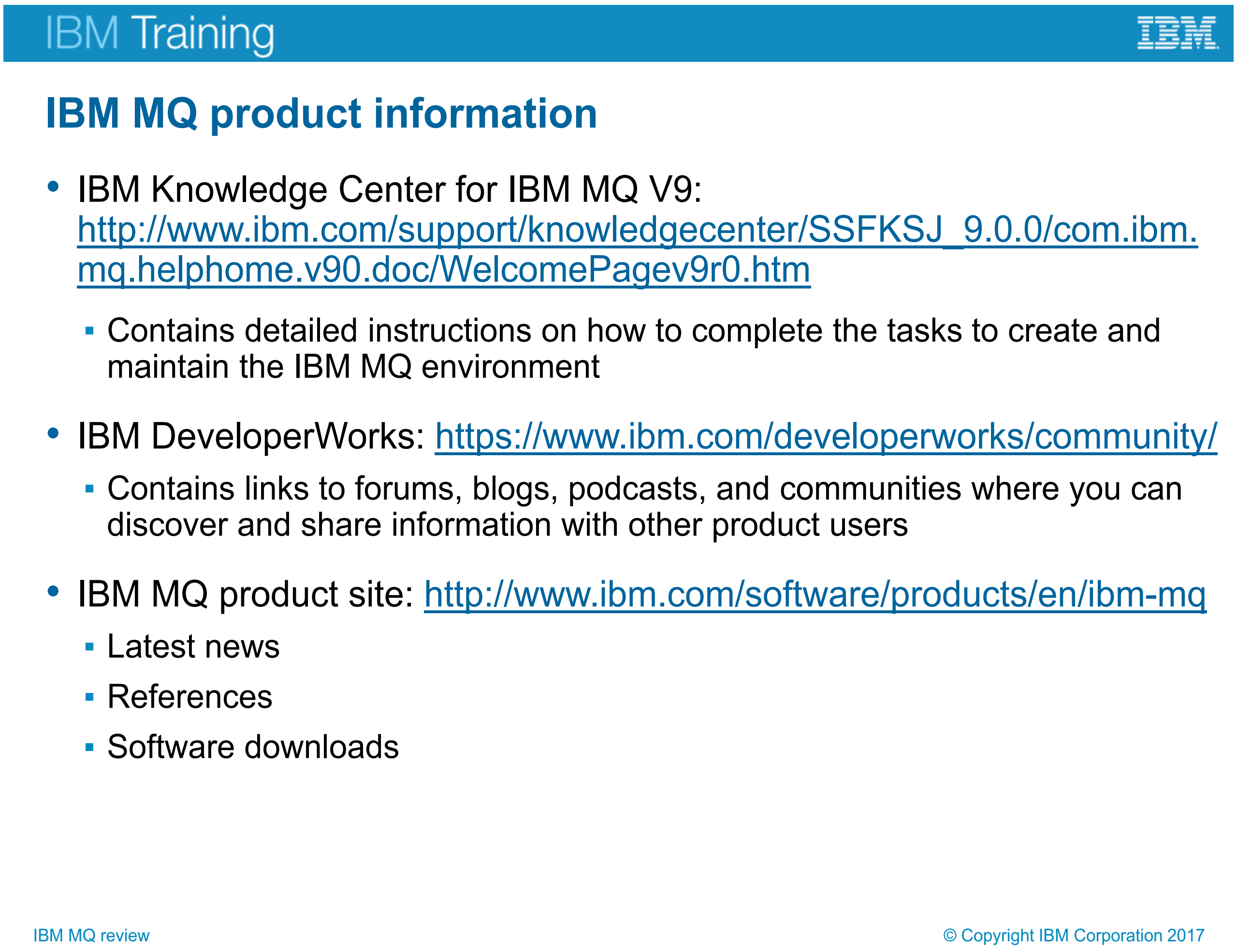
IBM MQ review © Copyright IBM Corporation 2017

Figure 1-21. IBM MQ Advanced

You can license the IBM MQ base software only. Optionally, you can license the IBM MQ Advanced package, which includes IBM MQ and three MQ extensions:

- IBM MQ Advanced Message Security provides advanced security features such as data encryption from the point it leaves the sending application to the point it enters the receiving application.
- IBM MQ Telemetry extends the universal messaging backbone with the MQTT protocol to a wide range of remote devices. It provides real-time access for enterprise applications to connect to a range of mobile devices, remote sensors, actuators, and other telemetry devices.
- IBM MQ Managed File Transfer facilitates the secure, reliable, and real-time transfer of files within the MQ network, and across a range of other platforms and networks. This manageable and auditable file transfer solution provides greater visibility and control of file transfer activity through a single system. IBM MQ Managed File Transfer is optimized to eliminate costly redundancies and lower maintenance costs, while it maximizes existing IT investments, to enable customers to move their business data stored in files over the IBM MQ infrastructure.

  IBM MQ Advanced helps to meet your current and emerging connectivity needs reliably with a complete, integrated, and universal messaging solution.



IBM Training

IBM MQ product information

- IBM Knowledge Center for IBM MQ V9: http://www.ibm.com/support/knowledgecenter/SSFKSJ_9.0.0/com.ibm.mq.helphome.v90.doc/WelcomePagev9r0.htm
  - Contains detailed instructions on how to complete the tasks to create and maintain the IBM MQ environment
- IBM DeveloperWorks: https://www.ibm.com/developerworks/community/
  - Contains links to forums, blogs, podcasts, and communities where you can discover and share information with other product users
- IBM MQ product site: http://www.ibm.com/software/products/en/ibm-mq
  - Latest news
  - References
  - Software downloads

IBM MQ review © Copyright IBM Corporation 2017

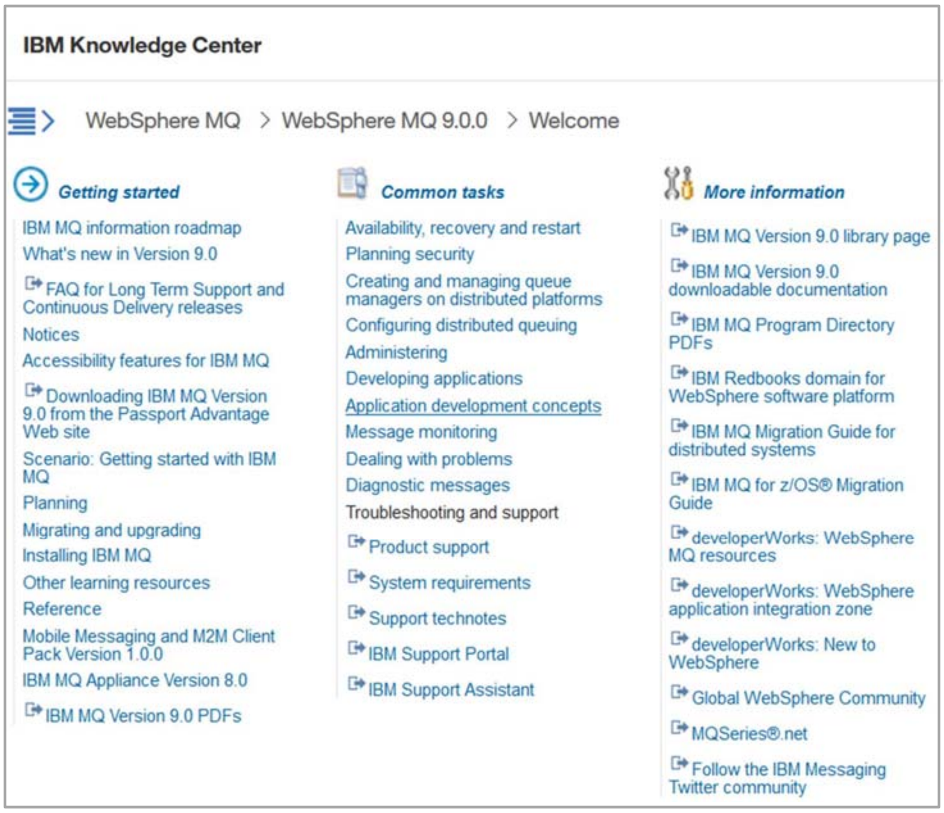
Figure 1-22. IBM MQ product information

The IBM Knowledge Center for IBM MQ V9 contains documentation for the MQ components that you install. In addition to reference documentation, the IBM MQ product documentation also contains links to tutorials and product tours.

Other resources, such as IBM DeveloperWorks and the IBM MQ product site, provide information about how to use new features, tips and techniques, and connections to user communities.

IBM Training

IBM Knowledge Center for IBM MQ V9



IBM Knowledge Center

WebSphere MQ > WebSphere MQ 9.0.0 > Welcome

Getting started

- IBM MQ information roadmap
- What's new in Version 9.0
- FAQ for Long Term Support and Continuous Delivery releases
- Notices
- Accessibility features for IBM MQ
- Downloading IBM MQ Version 9.0 from the Passport Advantage Web site
- Scenario: Getting started with IBM MQ
- Planning
- Migrating and upgrading
- Installing IBM MQ
- Other learning resources
- Reference
- Mobile Messaging and M2M Client Pack Version 1.0.0
- IBM MQ Appliance Version 8.0
- IBM MQ Version 9.0 PDFs

Common tasks

- Availability, recovery and restart
- Planning security
- Creating and managing queue managers on distributed platforms
- Configuring distributed queuing
- Administering
- Developing applications
- Application development concepts
- Message monitoring
- Dealing with problems
- Diagnostic messages
- Troubleshooting and support
- Product support
- System requirements
- Support technotes
- IBM Support Portal
- IBM Support Assistant

More information

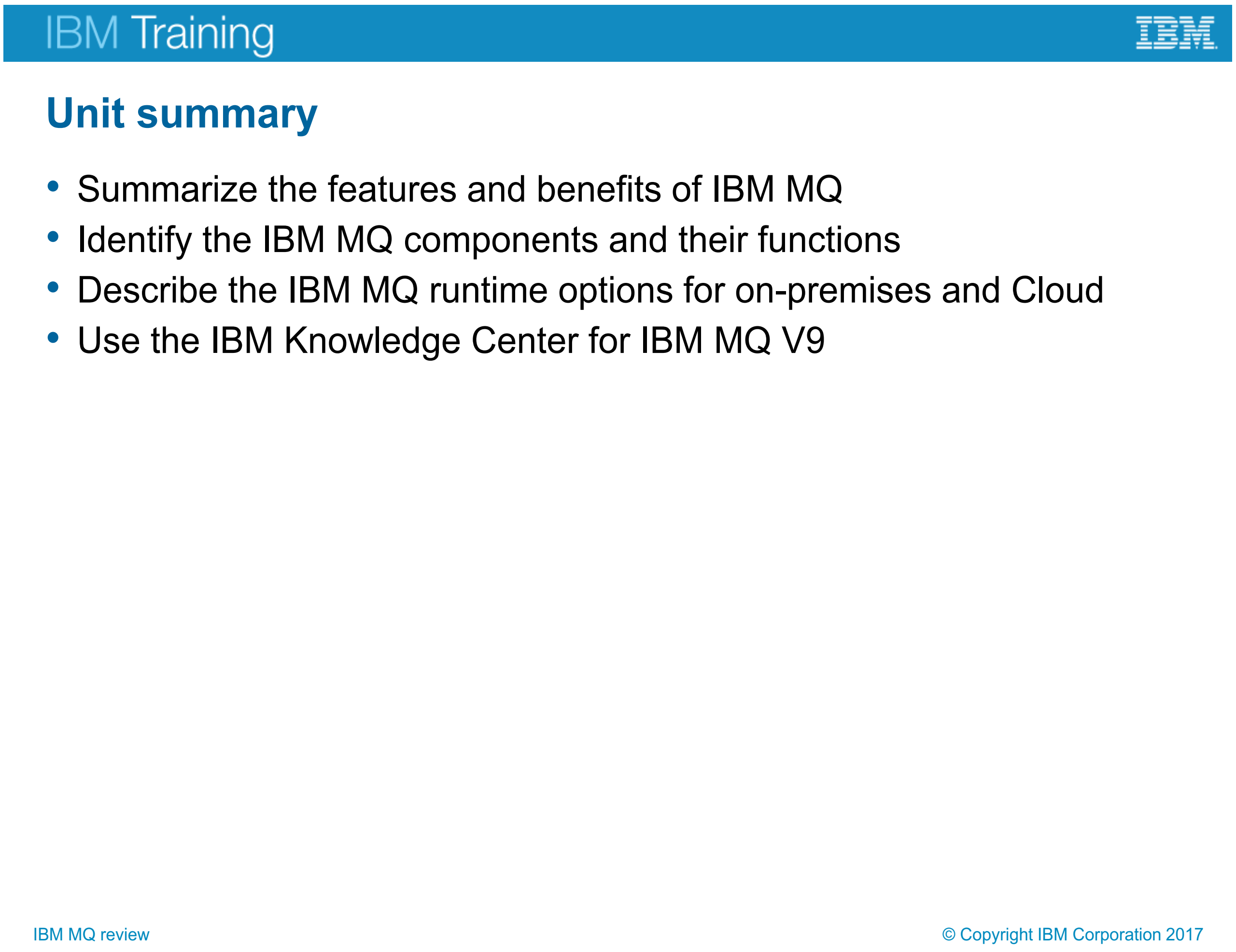
- IBM MQ Version 9.0 library page
- IBM MQ Version 9.0 downloadable documentation
- IBM MQ Program Directory PDFs
- IBM Redbooks domain for WebSphere software platform
- IBM MQ Migration Guide for distributed systems
- IBM MQ for z/OS® Migration Guide
- developerWorks: WebSphere MQ resources
- developerWorks: WebSphere application integration zone
- developerWorks: New to WebSphere
- Global WebSphere Community
- MQSeries®.net
- Follow the IBM Messaging Twitter community

- Primary source of information about IBM MQ
- Contains information to help you understand the product, and the ways in which you can use it to solve your business problems
- Downloadable on Windows and Linux

IBM MQ review © Copyright IBM Corporation 2017

Figure 1-23. IBM Knowledge Center for IBM MQ V9

The IBM Knowledge Center for IBM MQ v9 is the primary source of information about IBM MQ. It contains information to help you understand the product and describes implementation scenarios. The IBM Knowledge Center is available online and in download versions for Windows and Linux.



IBM Training

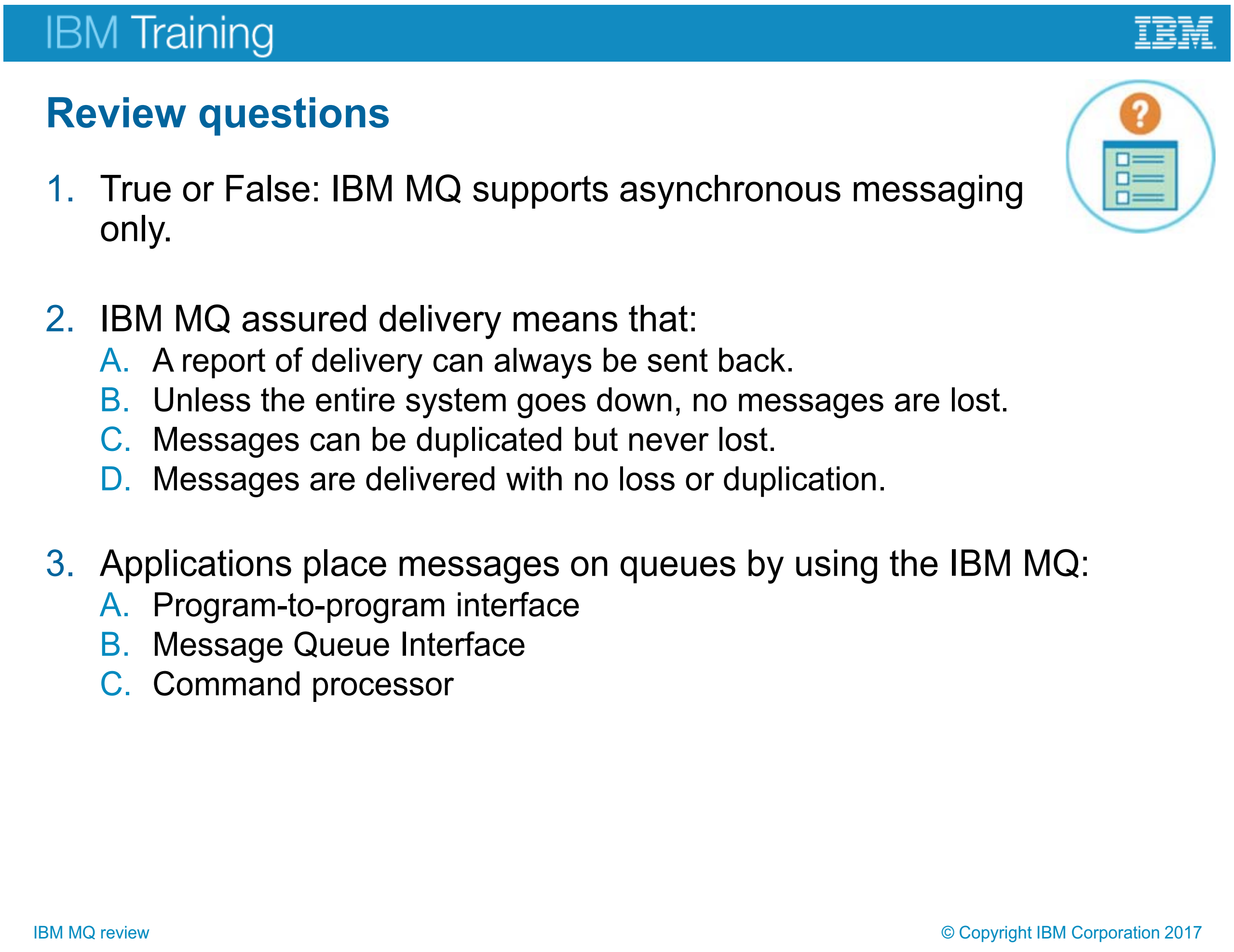
Unit summary

- Summarize the features and benefits of IBM MQ
- Identify the IBM MQ components and their functions
- Describe the IBM MQ runtime options for on-premises and Cloud
- Use the IBM Knowledge Center for IBM MQ V9

IBM MQ review

© Copyright IBM Corporation 2017

Figure 1-24. Unit summary



IBM Training

Review questions

1. True or False: IBM MQ supports asynchronous messaging only.

2. IBM MQ assured delivery means that:
   A. A report of delivery can always be sent back.
   B. Unless the entire system goes down, no messages are lost.
   C. Messages can be duplicated but never lost.
   D. Messages are delivered with no loss or duplication.

3. Applications place messages on queues by using the IBM MQ:
   A. Program-to-program interface
   B. Message Queue Interface
   C. Command processor

IBM MQ review

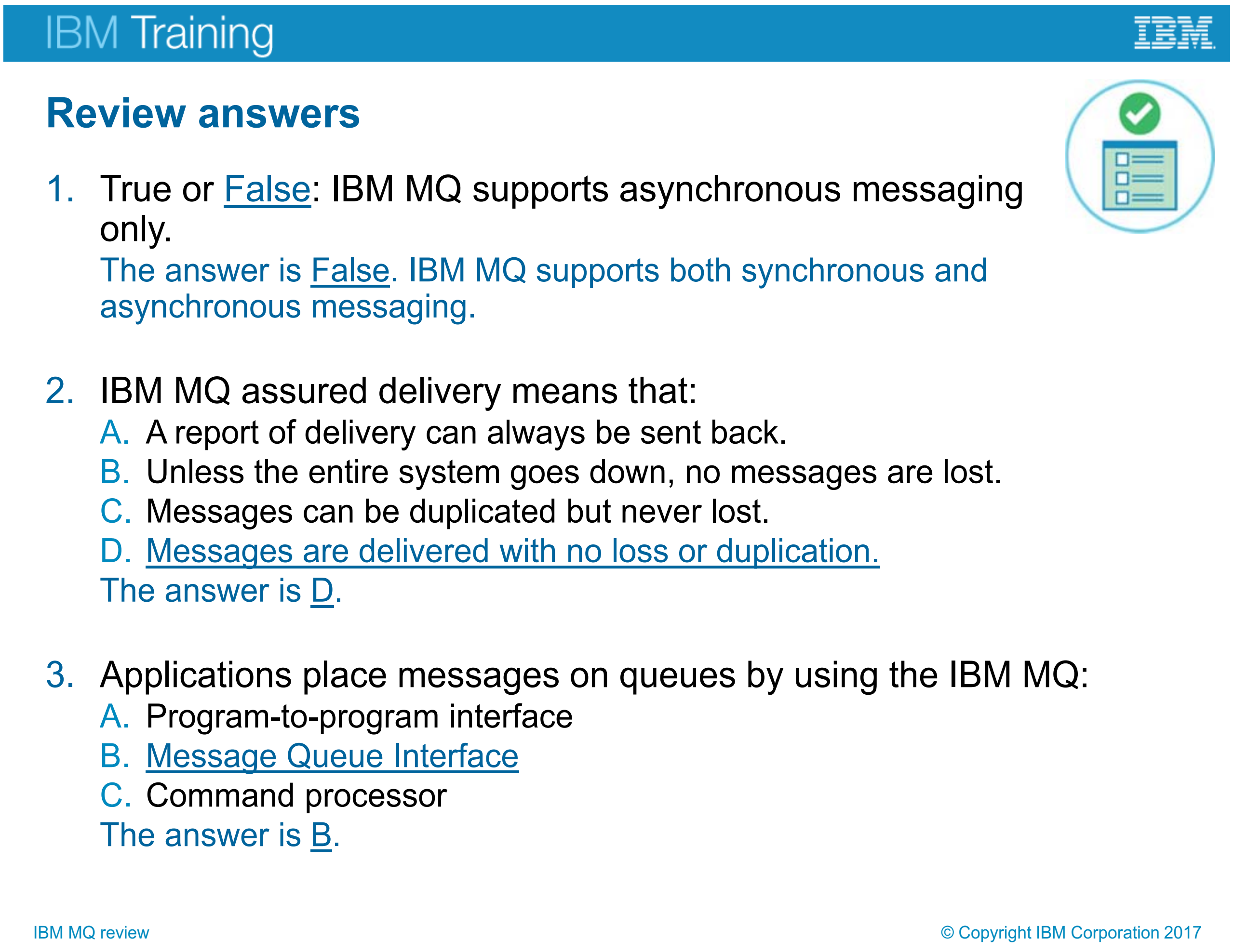
Figure 1-25. Review questions

Write your answers here:

1.

2.

3.



IBM Training

Review answers

1. True or False: IBM MQ supports asynchronous messaging only.
   The answer is False. IBM MQ supports both synchronous and asynchronous messaging.

2. IBM MQ assured delivery means that:
   A. A report of delivery can always be sent back.
   B. Unless the entire system goes down, no messages are lost.
   C. Messages can be duplicated but never lost.
   D. Messages are delivered with no loss or duplication.
   The answer is D.

3. Applications place messages on queues by using the IBM MQ:
   A. Program-to-program interface
   B. Message Queue Interface
   C. Command processor
   The answer is B.

IBM MQ review

Figure 1-26. Review answers

Unit 2. IBM MQ installation and deployment options

Estimated time

00:30

Overview

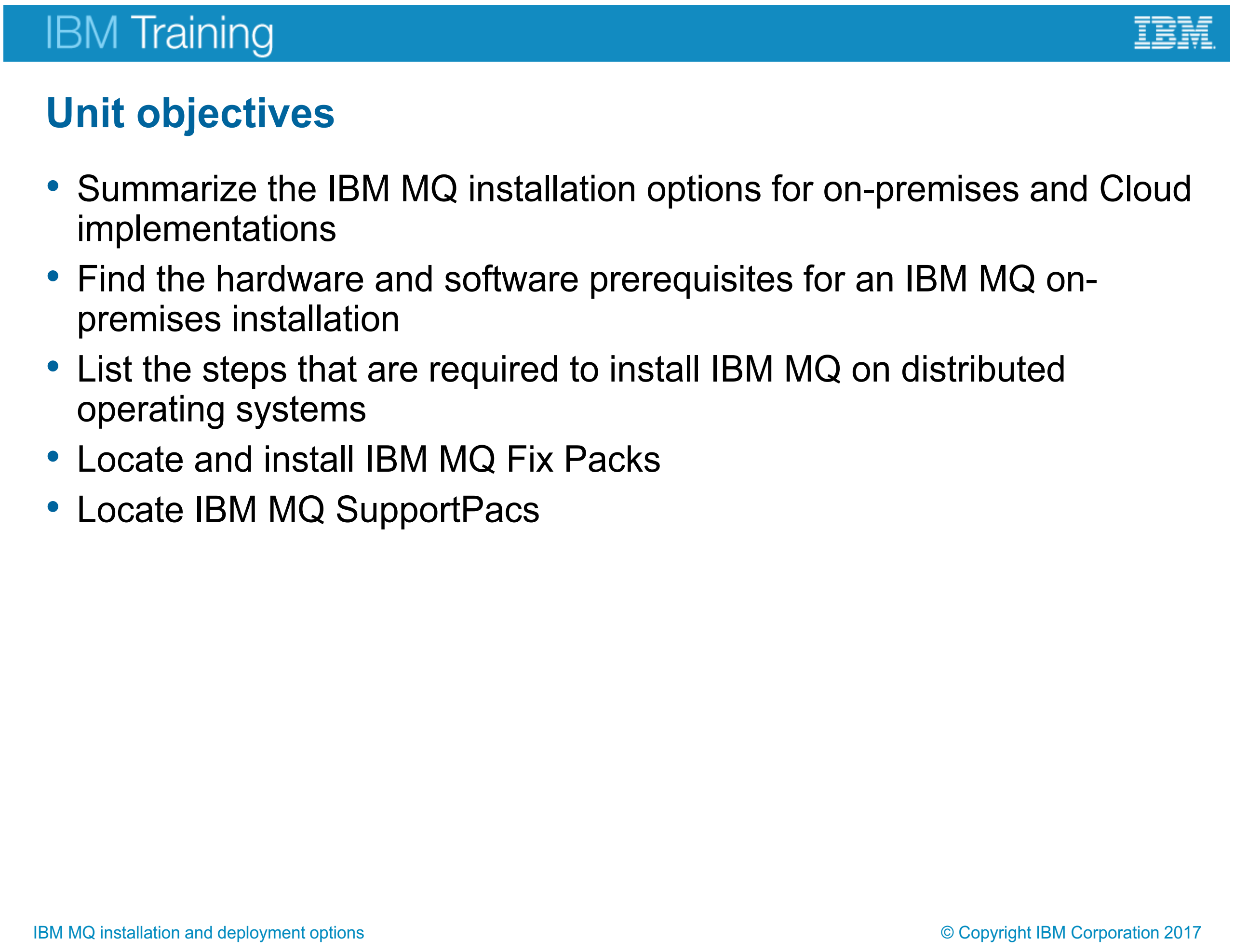
This unit describes the installation and deployment options for IBM MQ on-premises and in the Cloud. It also describes the release options for IBM MQ V9 and the process for locating and installing product Fix Packs.

How you will check your progress

- Review questions

References

IBM Knowledge Center for IBM MQ V9



IBM Training

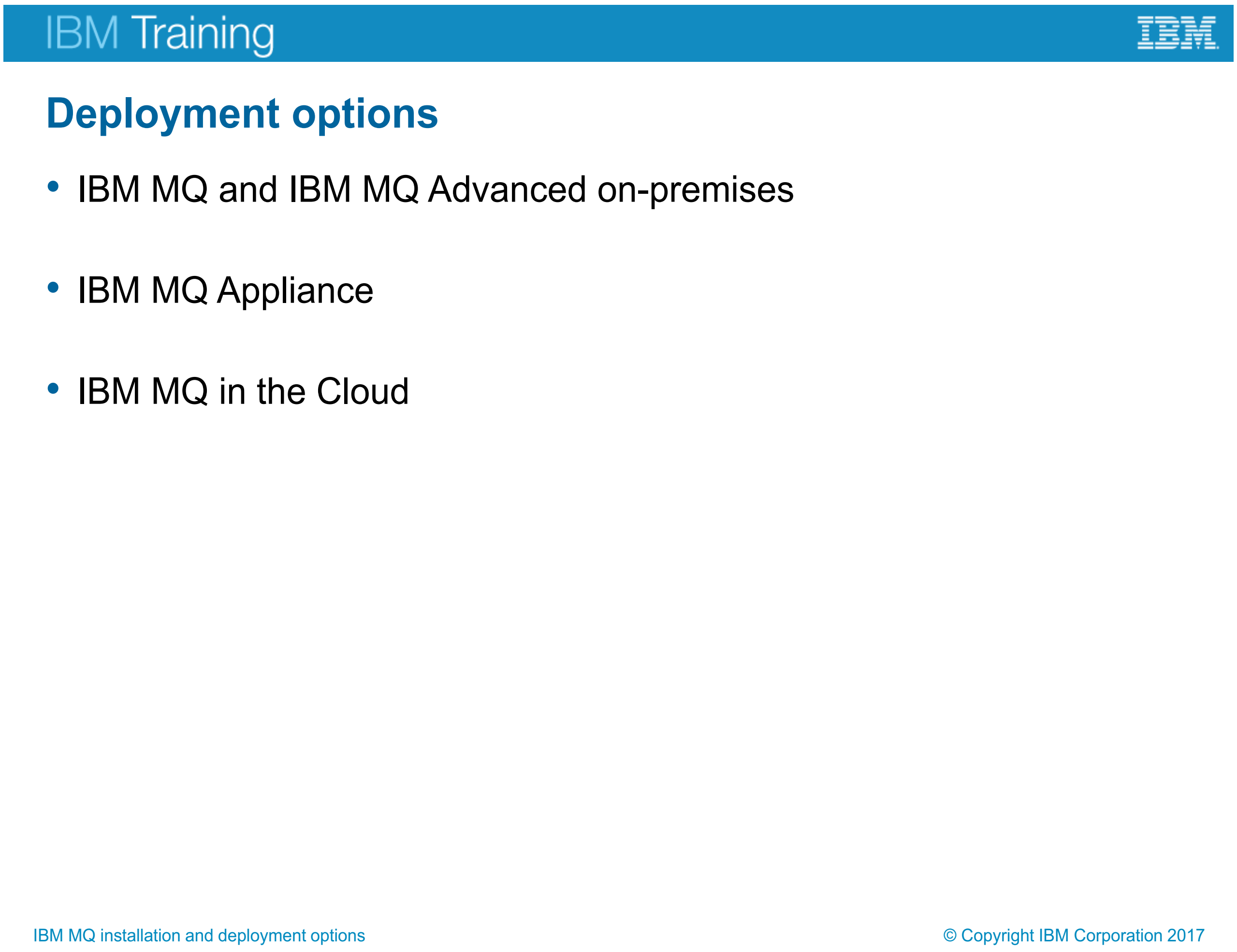
Unit objectives

- Summarize the IBM MQ installation options for on-premises and Cloud implementations
- Find the hardware and software prerequisites for an IBM MQ on-premises installation
- List the steps that are required to install IBM MQ on distributed operating systems
- Locate and install IBM MQ Fix Packs
- Locate IBM MQ SupportPacs

IBM MQ installation and deployment options

© Copyright IBM Corporation 2017

Figure 2-1. Unit objectives



IBM Training

Deployment options

- IBM MQ and IBM MQ Advanced on-premises
- IBM MQ Appliance
- IBM MQ in the Cloud

IBM MQ installation and deployment options

© Copyright IBM Corporation 2017

Figure 2-2. Deployment options

You have many options for deploying your IBM MQ solutions.

IBM MQ can be deployed in a traditional server network, on the IBM MQ Appliance, and in the Cloud.

IBM MQ and IBM MQ Advanced can be installed on a server network in your enterprise.

Optionally, you can use the IBM MQ Appliance, which provides an optimized version of MQ that runs in a hardware appliance. Offering MQ capability in hardware appliances provides more features and some appliance benefits.

You can extend IBM MQ to the Cloud by using IBM's PaaS (Platform as a Service) such as IBM PureApplication System, IBM Bluemix, IBM SoftLayer, and other Cloud and virtualized environments.

IBM Training

IBM MQ Appliance M2001

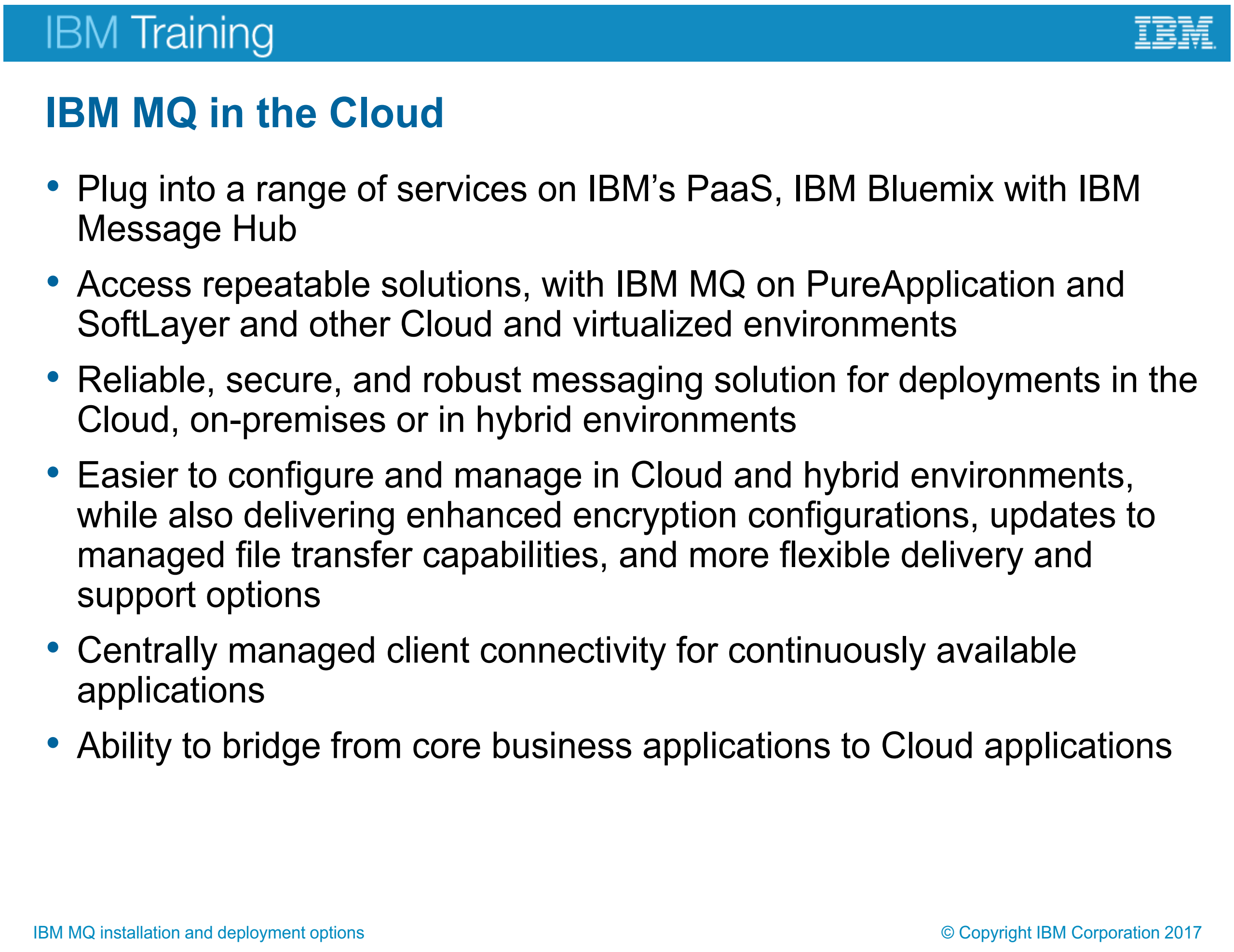
- Provides scalability and security of IBM MQ in the convenience, fast time-to-value, and low total cost of ownership of an appliance
  - Integrates seamlessly into IBM MQ networks
  - Familiar administration model for administrators with IBM MQ skills
  - Supports IBM MQ Light API
- Ideal for use as a messaging hub that runs queue managers that clients access, or to extend IBM MQ connectivity to a remote location
- Familiar feel for existing IBM MQ users such as application interfaces, administration, networking, clustering, and security
- Appliance-specific features such as built-in high availability and disaster recovery
- Helps extend applications to the Cloud as part of a hybrid infrastructure

IBM MQ installation and deployment options © Copyright IBM Corporation 2017

Figure 2-3. IBM MQ Appliance M2001

The IBM MQ Appliance provides the application connectivity performance of MQ software in a physical messaging appliance. It offers rapid deployment of enterprise messaging with easier administration. Performance and message throughput are optimized for the appliance's capability and configuration.

You can also use the MQ Appliance as the runtime messaging provider for applications that are written by using the IBM MQ Light API. It also supports connectivity from other programming interfaces such as the MQI and JMS.



IBM Training

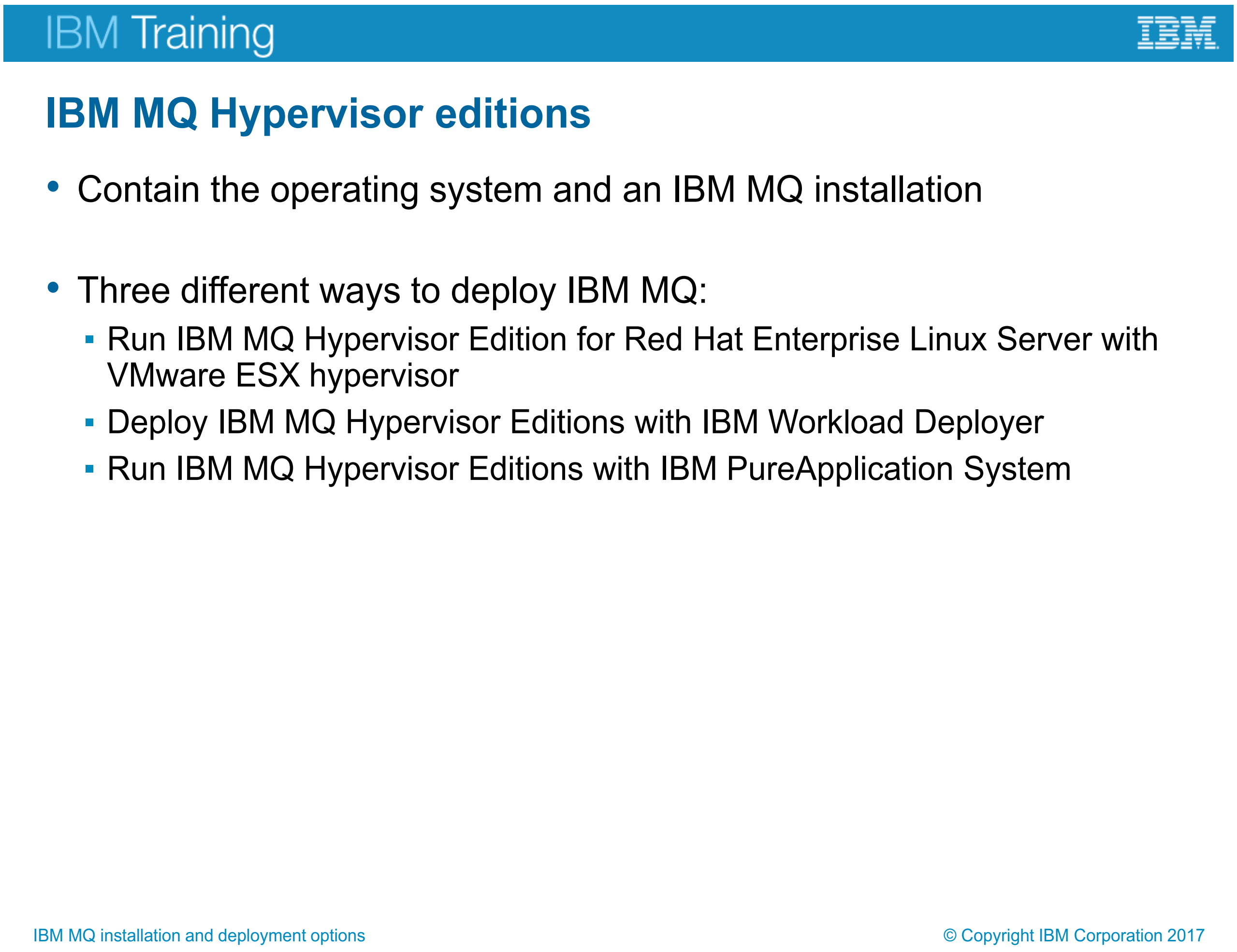
IBM MQ in the Cloud

- Plug into a range of services on IBM’s PaaS, IBM Bluemix with IBM Message Hub
- Access repeatable solutions, with IBM MQ on PureApplication and SoftLayer and other Cloud and virtualized environments
- Reliable, secure, and robust messaging solution for deployments in the Cloud, on-premises or in hybrid environments
- Easier to configure and manage in Cloud and hybrid environments, while also delivering enhanced encryption configurations, updates to managed file transfer capabilities, and more flexible delivery and support options
- Centrally managed client connectivity for continuously available applications
- Ability to bridge from core business applications to Cloud applications

IBM MQ installation and deployment options © Copyright IBM Corporation 2017

Figure 2-4. IBM MQ in the Cloud

You can extend IBM MQ to the Cloud by using IBM PureApp, SoftLayer, and IBM Bluemix.



IBM Training

IBM MQ Hypervisor editions

- Contain the operating system and an IBM MQ installation
- Three different ways to deploy IBM MQ:
  - Run IBM MQ Hypervisor Edition for Red Hat Enterprise Linux Server with VMware ESX hypervisor
  - Deploy IBM MQ Hypervisor Editions with IBM Workload Deployer
  - Run IBM MQ Hypervisor Editions with IBM PureApplication System

IBM MQ installation and deployment options

© Copyright IBM Corporation 2017

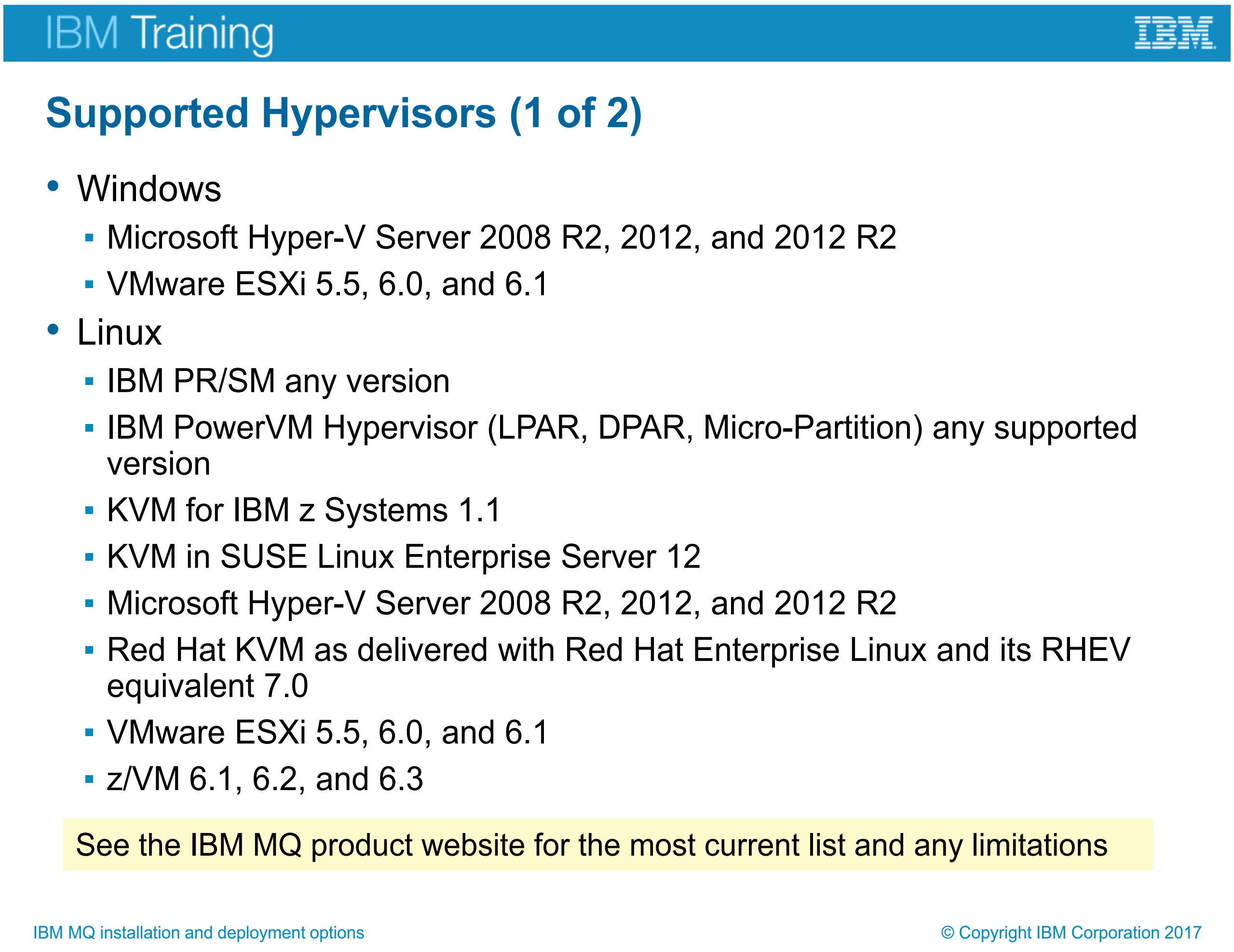
Figure 2-5. IBM MQ Hypervisor editions

Hypervisors are software or firmware components that can virtualize system resources. Examples of hypervisors are PowerVM for AIX and VMware ESXServer.

The MQ Hypervisor Editions are self-contained virtual machine images. The images contain the operating system and MQ. You can deploy the virtual machine images into a Cloud with IBM Workload Deployer or IBM PureApplication System.

The other resources include an MQ basic part, script packages, and a Python script. The Python script loads the MQ virtual image and script packages onto an appliance, and creates a default MQ virtual system pattern.

With the cluster script packages, you can configure a pattern to add or remove a cluster of queue managers. The other script package runs the MQSC command tool.



IBM Training

Supported Hypervisors (1 of 2)

- Windows
  - Microsoft Hyper-V Server 2008 R2, 2012, and 2012 R2
  - VMware ESXi 5.5, 6.0, and 6.1
- Linux
  - IBM PR/SM any version
  - IBM PowerVM Hypervisor (LPAR, DPAR, Micro-Partition) any supported version
  - KVM for IBM z Systems 1.1
  - KVM in SUSE Linux Enterprise Server 12
  - Microsoft Hyper-V Server 2008 R2, 2012, and 2012 R2
  - Red Hat KVM as delivered with Red Hat Enterprise Linux and its RHEV equivalent 7.0
  - VMware ESXi 5.5, 6.0, and 6.1
  - z/VM 6.1, 6.2, and 6.3

See the IBM MQ product website for the most current list and any limitations

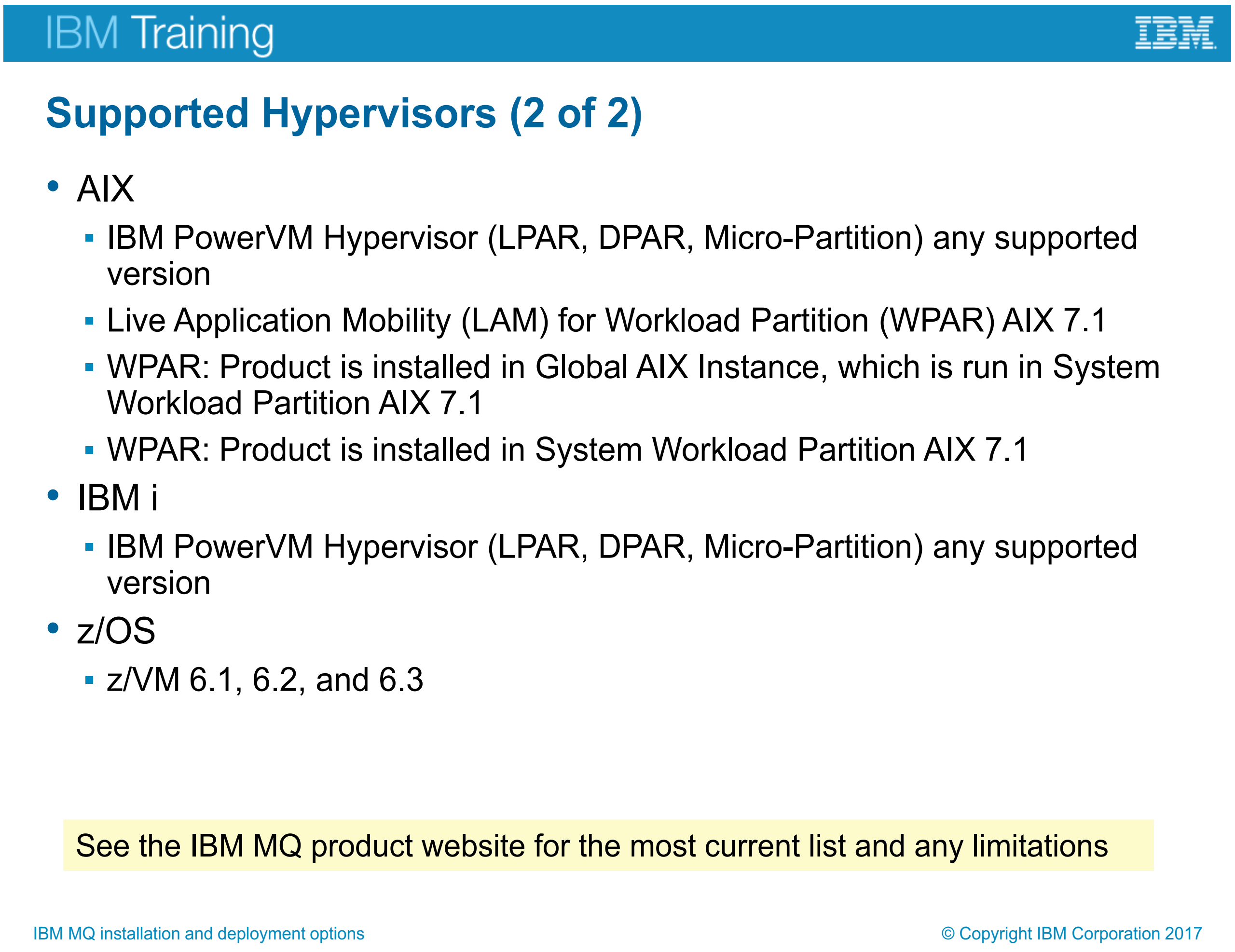
IBM MQ installation and deployment options

© Copyright IBM Corporation 2017

Figure 2-6. Supported Hypervisors (1 of 2)

This figure summarizes the hypervisors that support IBM MQ V9 on Windows and Linux.

See the IBM MQ product website for the most current list of hypervisors for IBM MQ V9.



IBM Training

Supported Hypervisors (2 of 2)

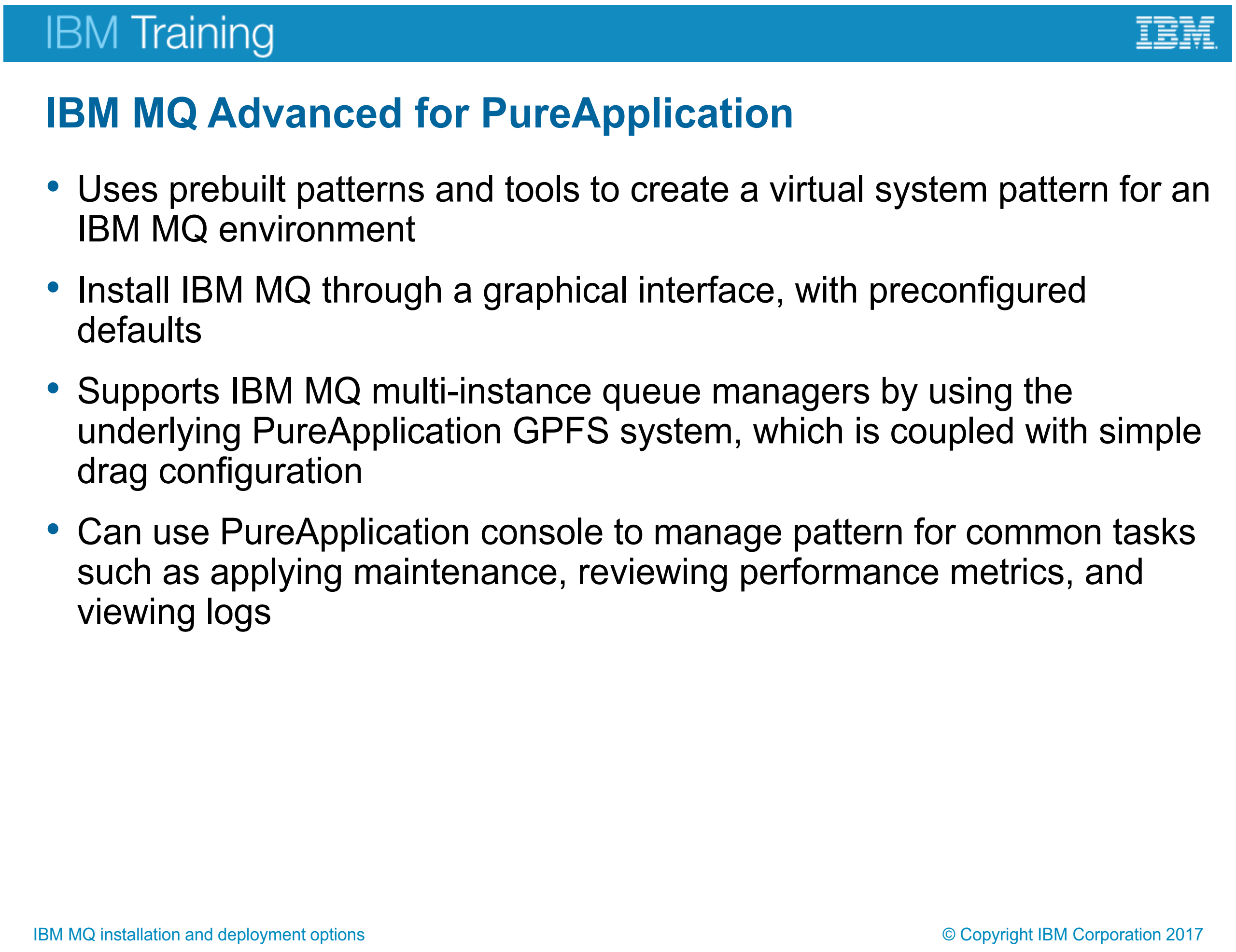
- AIX
  - IBM PowerVM Hypervisor (LPAR, DPAR, Micro-Partition) any supported version
  - Live Application Mobility (LAM) for Workload Partition (WPAR) AIX 7.1
  - WPAR: Product is installed in Global AIX Instance, which is run in System Workload Partition AIX 7.1
  - WPAR: Product is installed in System Workload Partition AIX 7.1
- IBM i
  - IBM PowerVM Hypervisor (LPAR, DPAR, Micro-Partition) any supported version
- z/OS
  - z/VM 6.1, 6.2, and 6.3

See the IBM MQ product website for the most current list and any limitations

IBM MQ installation and deployment options © Copyright IBM Corporation 2017

Figure 2-7. Supported Hypervisors (2 of 2)

This figure lists hypervisors that are available for IBM MQ on AIX, IBM i, and z/OS.



IBM Training

IBM MQ Advanced for PureApplication

- Uses prebuilt patterns and tools to create a virtual system pattern for an IBM MQ environment
- Install IBM MQ through a graphical interface, with preconfigured defaults
- Supports IBM MQ multi-instance queue managers by using the underlying PureApplication GPFS system, which is coupled with simple drag configuration
- Can use PureApplication console to manage pattern for common tasks such as applying maintenance, reviewing performance metrics, and viewing logs

IBM MQ installation and deployment options

© Copyright IBM Corporation 2017

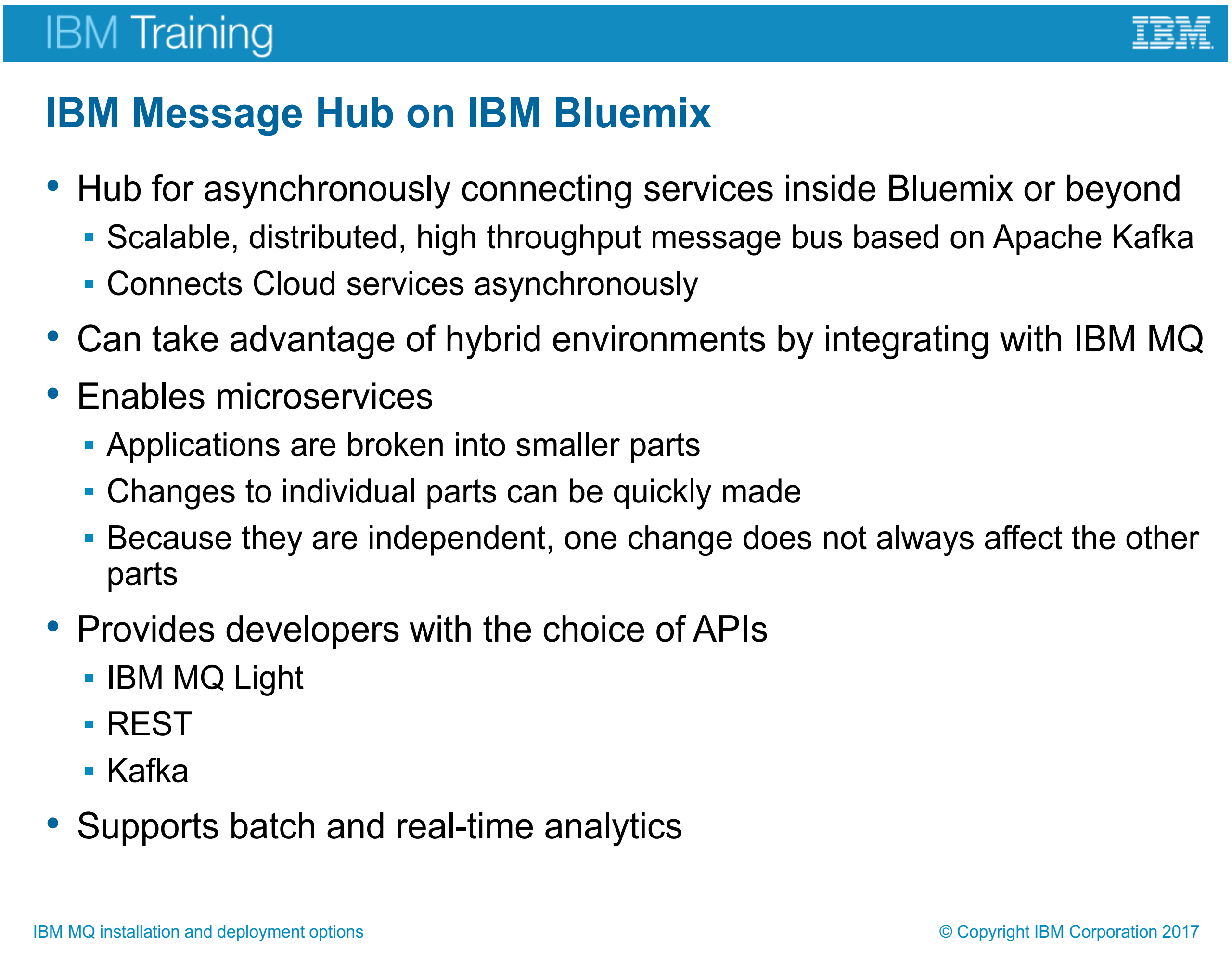
Figure 2-8. IBM MQ Advanced for PureApplication

IBM MQ Advanced for PureApplication uses prebuilt patterns and tools to create a virtual system pattern for an MQ environment.

With IBM MQ Advanced for PureApplication, you install MQ through a graphical interface with preconfigured defaults; it is not necessary to use any commands to install MQ.

The MQ Advanced PureApplication pattern supports MQ multi-instance queue managers by using the underlying PureApplication GPFS system that is coupled with simple configuration.

The PureApplication console can manage the MQ Advanced PureApplication pattern for common tasks such as applying maintenance, reviewing performance metrics, and viewing logs.



IBM Training

IBM Message Hub on IBM Bluemix

- Hub for asynchronously connecting services inside Bluemix or beyond
  - Scalable, distributed, high throughput message bus based on Apache Kafka
  - Connects Cloud services asynchronously
- Can take advantage of hybrid environments by integrating with IBM MQ
- Enables microservices
  - Applications are broken into smaller parts
  - Changes to individual parts can be quickly made
  - Because they are independent, one change does not always affect the other parts
- Provides developers with the choice of APIs
  - IBM MQ Light
  - REST
  - Kafka
- Supports batch and real-time analytics

IBM MQ installation and deployment options

© Copyright IBM Corporation 2017

Figure 2-9. IBM Message Hub on IBM Bluemix

With the support of the MQ Light API, it is possible to connect MQ to IBM Bluemix so that you can take advantage of the benefits of working in the Cloud.

MQ connects to IBM Message Hub, which is a fast, scalable, durable service that is based on Apache Kafka. Message Hub sits between services in Bluemix and provides buffering, load balancing, and error handling. It is designed to work easily with a range of services that enable further simplification.

With the advent of Cloud, you have access to more endpoints than ever before. This movement towards Cloud usage prompted more interest in analysis of data. Message Hub can use data from one or many endpoints, which include MQ, and feed to one or more analytics engines, allowing different types of analysis to be run on the same data.

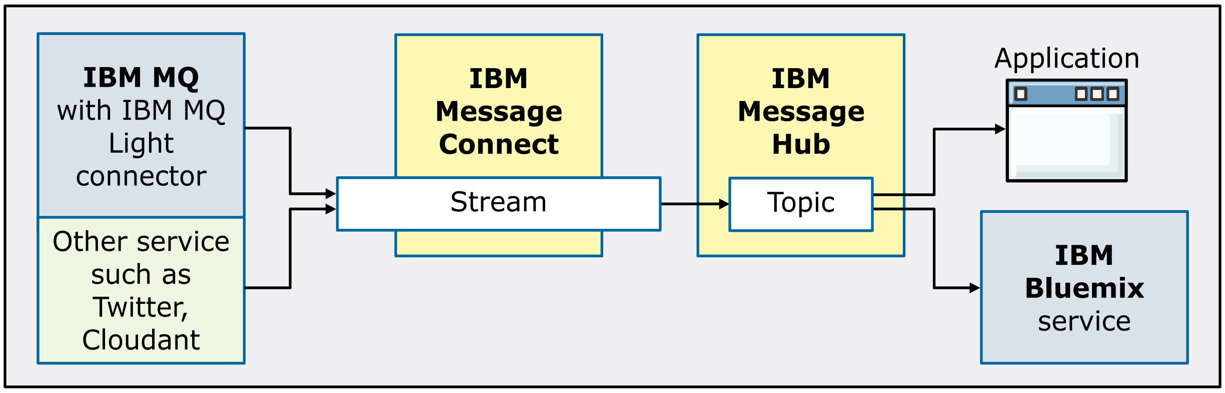
Message Hub enables development in a microservices framework. Microservices accelerate integration to help you use applications and data to respond quickly to business demands and create innovative solutions. For example, with Message Hub you can stream batch and real-time data to analytics applications to gain greater insight and advantage from your information.

Message Hub provides messaging for services inside and outside of Bluemix, and can also be used to take advantage of hybrid environments by integrating with MQ.

IBM Training

IBM Message Connect on IBM Bluemix

- Connects enterprise IBM MQ message network to the Cloud
  - Scalable, distributed, high throughput message bus
  - Loads data into analytics services to optimize business decisions
  - Builds on IBM MQ's dominance in enterprise messaging
- Combination of Message Hub and Message Connect provide features similar to other publish/subscribe systems
  - At one end, one or more publishers write data on to a stream
  - At the other end, one or more consumers receive a push notification of the event and the data packets from the publisher



IBM MQ installation and deployment options

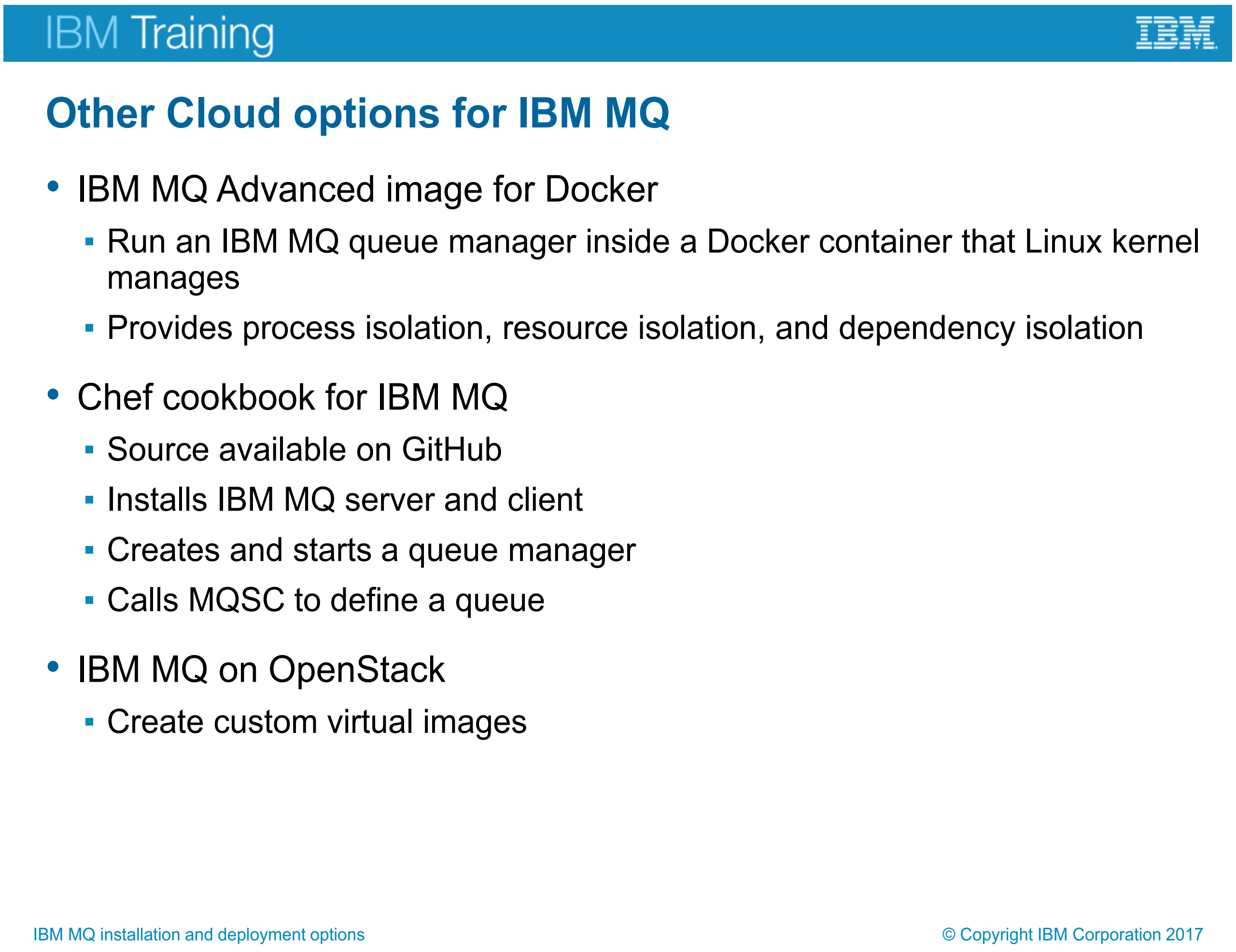
Figure 2-10. IBM Message Connect on IBM Bluemix

IBM Message Connect connects streams of events from various sources and feeds them to your applications and services in Bluemix by using the IBM Message Hub publish/subscribe messaging service.

Events travel along streams, which act as pathways for data from event sources into Message Hub.

Connectors are prebuilt to connect to data sources and flow-specific data points onto a stream. For example, a connector can provide connectivity to Twitter's streaming search API. Tweets are channeled from Twitter to your stream without multiple connections to Twitter itself. As a Message Connect user, all you need to do is supply your Twitter authentication credentials and search criteria. Message Connect takes care of the rest, and you do not have to write a single line of code.

You can configure Bluemix applications to provide processing for the data, or to act based on incoming messages.



IBM Training

Other Cloud options for IBM MQ

- IBM MQ Advanced image for Docker
  - Run an IBM MQ queue manager inside a Docker container that Linux kernel manages
  - Provides process isolation, resource isolation, and dependency isolation
- Chef cookbook for IBM MQ
  - Source available on GitHub
  - Installs IBM MQ server and client
  - Creates and starts a queue manager
  - Calls MQSC to define a queue
- IBM MQ on OpenStack
  - Create custom virtual images

IBM MQ installation and deployment options © Copyright IBM Corporation 2017

Figure 2-11. Other Cloud options for IBM MQ

You can also choose to run MQ in Docker or OpenStack.

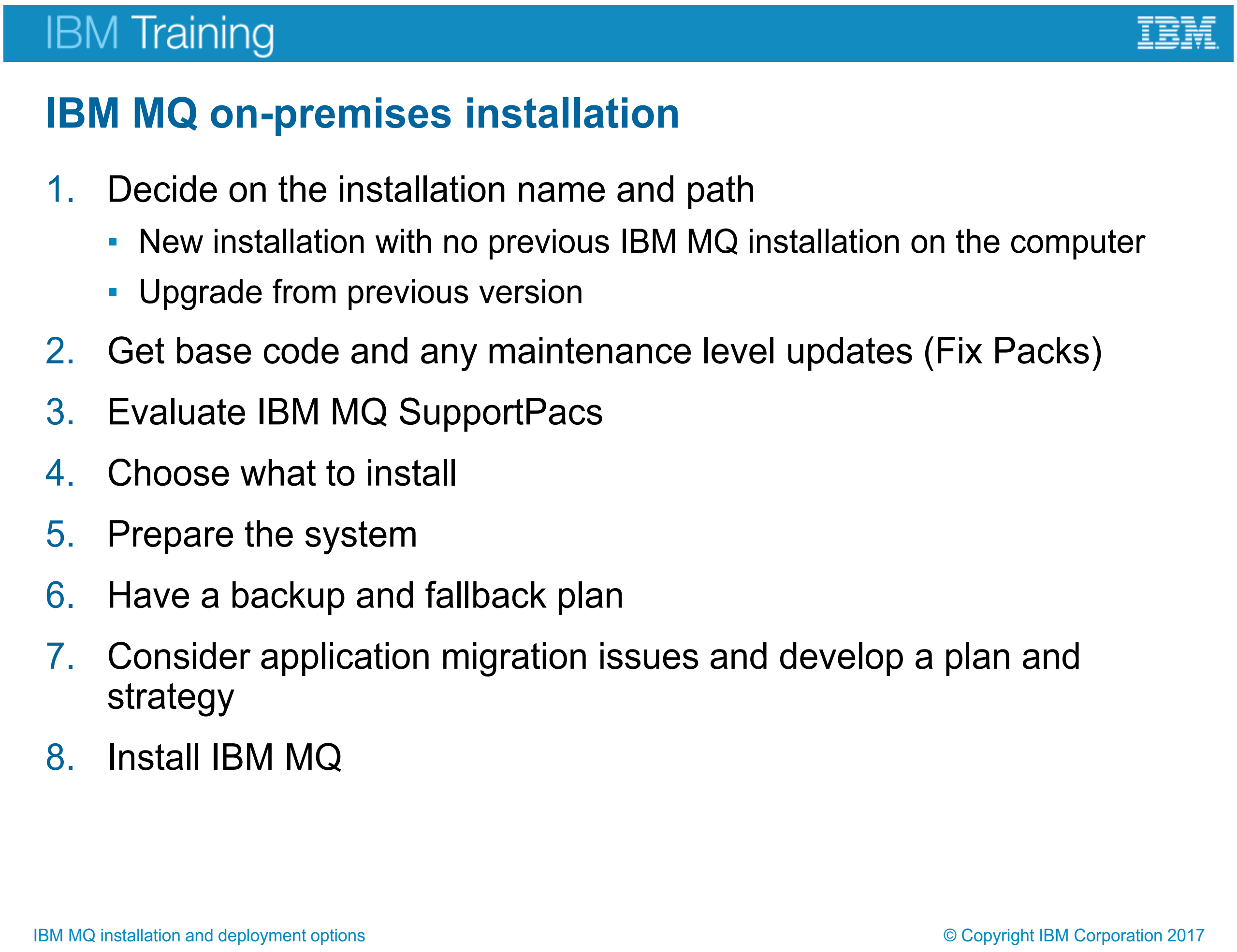
With the IBM MQ Advanced image for Docker, you can run an MQ queue manager inside a Docker container, which can be useful for several reasons:

- All the processes that are associated with MQ are run in their own process space.
- You can limit the amount of memory and CPU that you allocate to a container.
- All the software on which MQ depends is included in the MQ image.

OpenStack is an infrastructure as a service (IaaS) offering, which provides a platform and unified API for managing infrastructure resources. These infrastructure resources include virtual machines, networks, and storage. MQ currently declares support for many of the individual OpenStack provider technologies, such as the KVM, Hyper-V, and VMware hypervisors.

MQ is also available as an image on Amazon Web Services and on Microsoft Azure.

GitHub contains Chef Cookbook for a basic IBM MQ installation.



IBM Training

IBM MQ on-premises installation

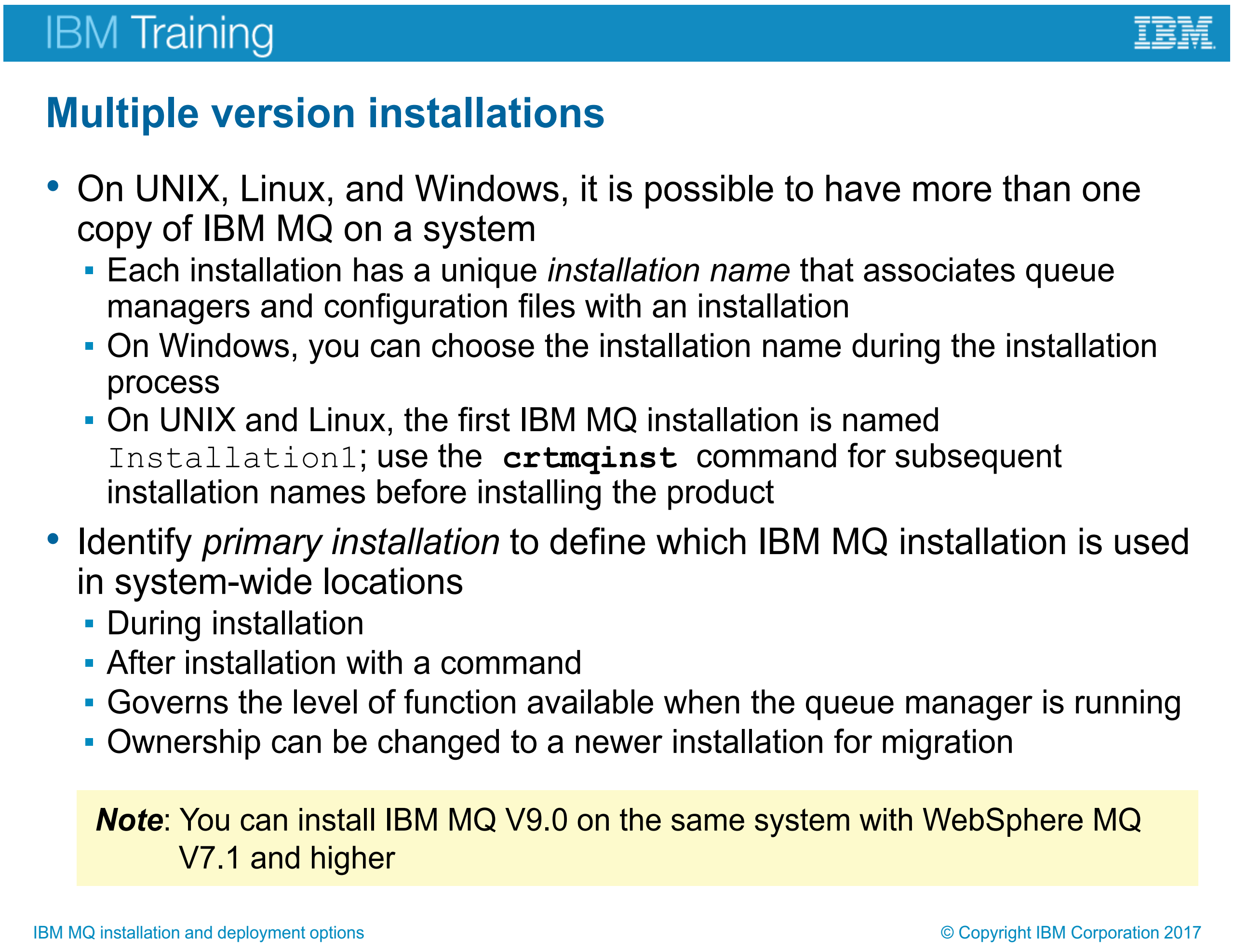
1. Decide on the installation name and path
   - New installation with no previous IBM MQ installation on the computer
   - Upgrade from previous version
2. Get base code and any maintenance level updates (Fix Packs)
3. Evaluate IBM MQ SupportPacs
4. Choose what to install
5. Prepare the system
6. Have a backup and fallback plan
7. Consider application migration issues and develop a plan and strategy
8. Install IBM MQ

IBM MQ installation and deployment options

© Copyright IBM Corporation 2017

Figure 2-12. IBM MQ on-premises installation

This figure summarizes the tasks for an IBM MQ on-premises installation. Each of these steps is described in more detail in this unit.



IBM Training

Multiple version installations

- On UNIX, Linux, and Windows, it is possible to have more than one copy of IBM MQ on a system
  - Each installation has a unique *installation name* that associates queue managers and configuration files with an installation
  - On Windows, you can choose the installation name during the installation process
  - On UNIX and Linux, the first IBM MQ installation is named `Installation1`; use the **`crtmqinst`** command for subsequent installation names before installing the product
- Identify *primary installation* to define which IBM MQ installation is used in system-wide locations
  - During installation
  - After installation with a command
  - Governs the level of function available when the queue manager is running
  - Ownership can be changed to a newer installation for migration

Note: You can install IBM MQ V9.0 on the same system with WebSphere MQ V7.1 and higher

IBM MQ installation and deployment options © Copyright IBM Corporation 2017

Figure 2-13. Multiple version installations

You can install more than one copy of IBM MQ on distributed operating systems. An installation is the collection of binary libraries that make up MQ.

Multiple version installation support simplifies migration strategies because you can continue to use one version of MQ and gradually upgrade applications to a new version, without needing parallel hardware. In addition, vendor-acquired applications can embed MQ in a private directory, and can choose which versions of MQ are supported, without worrying about the “visible” version of MQ that a user might be using.

The installation details vary by operating system, but conceptually you first define an installation, and give it a name and description. You then run the installation program, which copies the code from media onto the disk in your chosen directories.

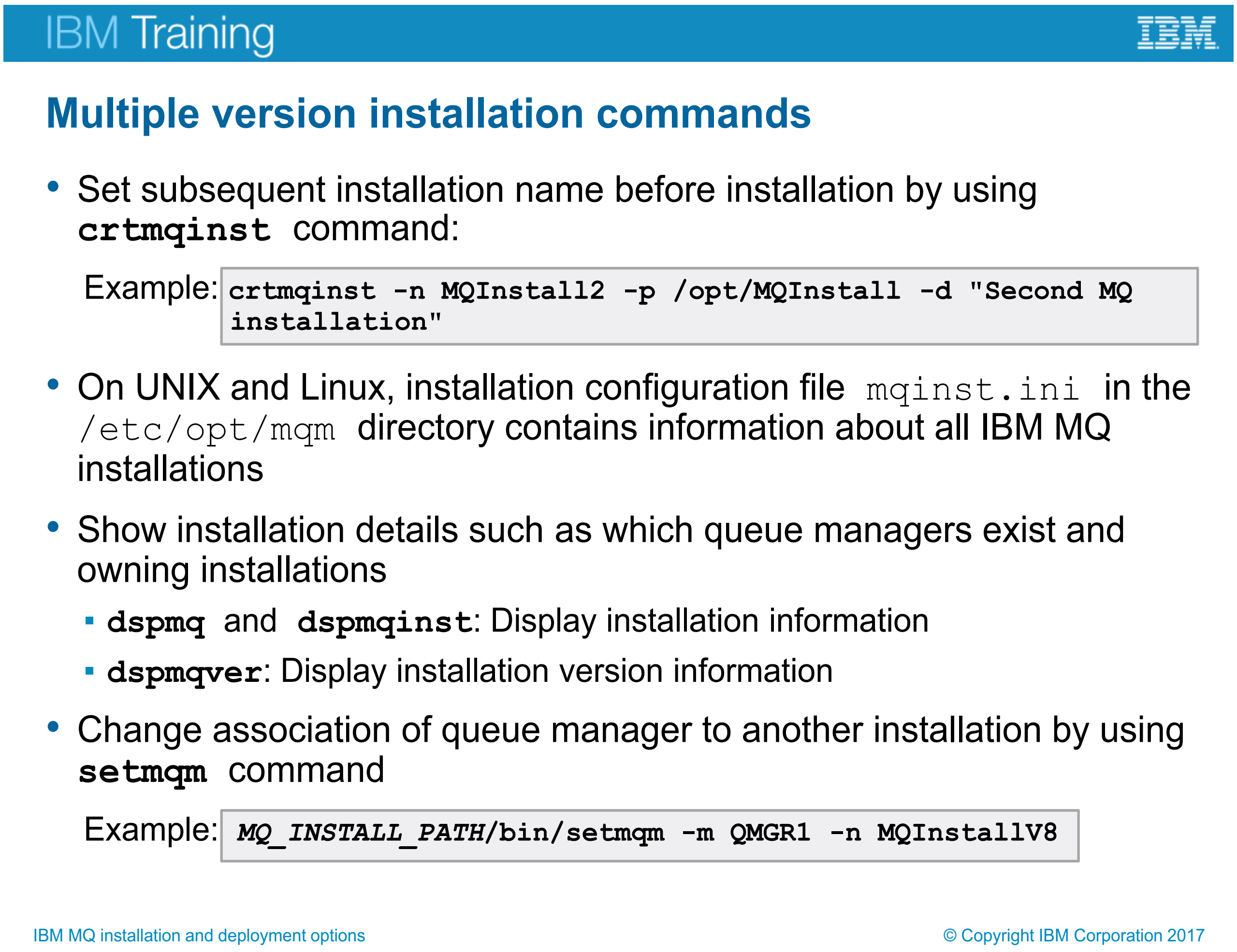
Generally, no “default” paths are created to an installation. However, a primary installation does insert itself into default locations.

You can have a single primary installation only on a system. On Windows, one installation is always identified as the primary installation because some operating system functions, such as how Windows **`.dll`** files are registered, require a single location.

An installation owns each queue manager, which governs the level of function available when a queue manager runs. When you install a newer level of code, the queue manager can be moved to that newer installation, and new functions can be used.

For easier migration, you can have an existing copy of WebSphere MQ V7.0.1.6 (or higher) on your systems. It is not necessary to upgrade before you use the multiple installation capabilities. If you already have WebSphere MQ V7.0.1 installed on your system, it is always the primary installation on all operating systems.

When you install IBM MQ, you can choose the directory into which it is installed.



IBM Training

Multiple version installation commands

- Set subsequent installation name before installation by using **`crtmqinst`** command:

  Example:
  ```
  crtmqinst -n MQInstall2 -p /opt/MQInstall -d "Second MQ
  installation"
  ```
- On UNIX and Linux, installation configuration file `mqinst.ini` in the `/etc/opt/mqm` directory contains information about all IBM MQ installations
- Show installation details such as which queue managers exist and owning installations
  - **`dspmq`** and **`dspmqinst`**: Display installation information
  - **`dspmqver`**: Display installation version information
- Change association of queue manager to another installation by using **`setmqm`** command

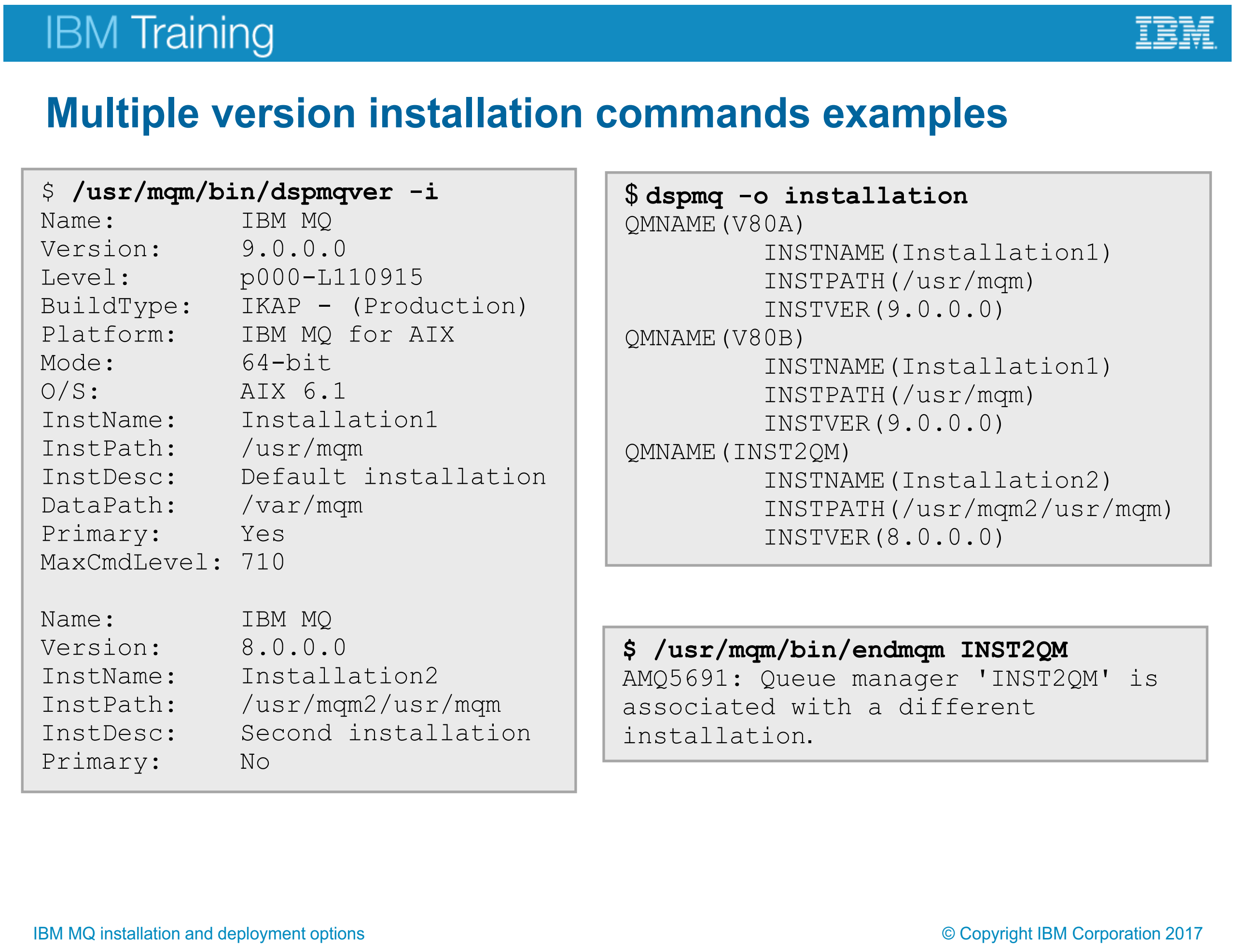
  Example:
  ```
  MQ_INSTALL_PATH/bin/setmqm -m QMGR1 -n MQInstallV8
  ```

IBM MQ installation and deployment options © Copyright IBM Corporation 2017

Figure 2-14. Multiple version installation commands

IBM MQ commands can be used to examine and verify the installation information. In addition, installation details are maintained in the `/etc/opt/mqm/mqinst.ini` file on UNIX and Linux, and in the registry on Windows.

The commands are not in the default PATH for UNIX and Linux. If you are not working with the primary installation, all the control commands must include an explicit path. You can also use the **setmqenv** command to automatically set up the environment for use with a different installation of MQ. The **setmqenv** command and multiple version installations are described in detail in the IBM MQ product documentation.



IBM Training

Multiple version installation commands examples

```
$ /usr/mqm/bin/dspmqver -i
Name:          IBM MQ
Version:       9.0.0.0
Level:         p000-L110915
BuildType:     IKAP - (Production)
Platform:      IBM MQ for AIX
Mode:          64-bit
O/S:           AIX 6.1
InstName:      Installation1
InstPath:      /usr/mqm
InstDesc:      Default installation
DataPath:      /var/mqm
Primary:       Yes
MaxCmdLevel:   710

Name:          IBM MQ
Version:       8.0.0.0
InstName:      Installation2
InstPath:      /usr/mqm2/usr/mqm
InstDesc:      Second installation
Primary:       No
```

```
$ dspmq -o installation
QMNAME(V80A)
         INSTNAME(Installation1)
         INSTPATH(/usr/mqm)
         INSTVER(9.0.0.0)
QMNAME(V80B)
         INSTNAME(Installation1)
         INSTPATH(/usr/mqm)
         INSTVER(9.0.0.0)
QMNAME(INST2QM)
         INSTNAME(Installation2)
         INSTPATH(/usr/mqm2/usr/mqm)
         INSTVER(8.0.0.0)
```

```
$ /usr/mqm/bin/endmqm INST2QM
AMQ5691: Queue manager 'INST2QM' is
associated with a different
installation.
```

IBM MQ installation and deployment options

© Copyright IBM Corporation 2017

Figure 2-15. Multiple version installation commands examples

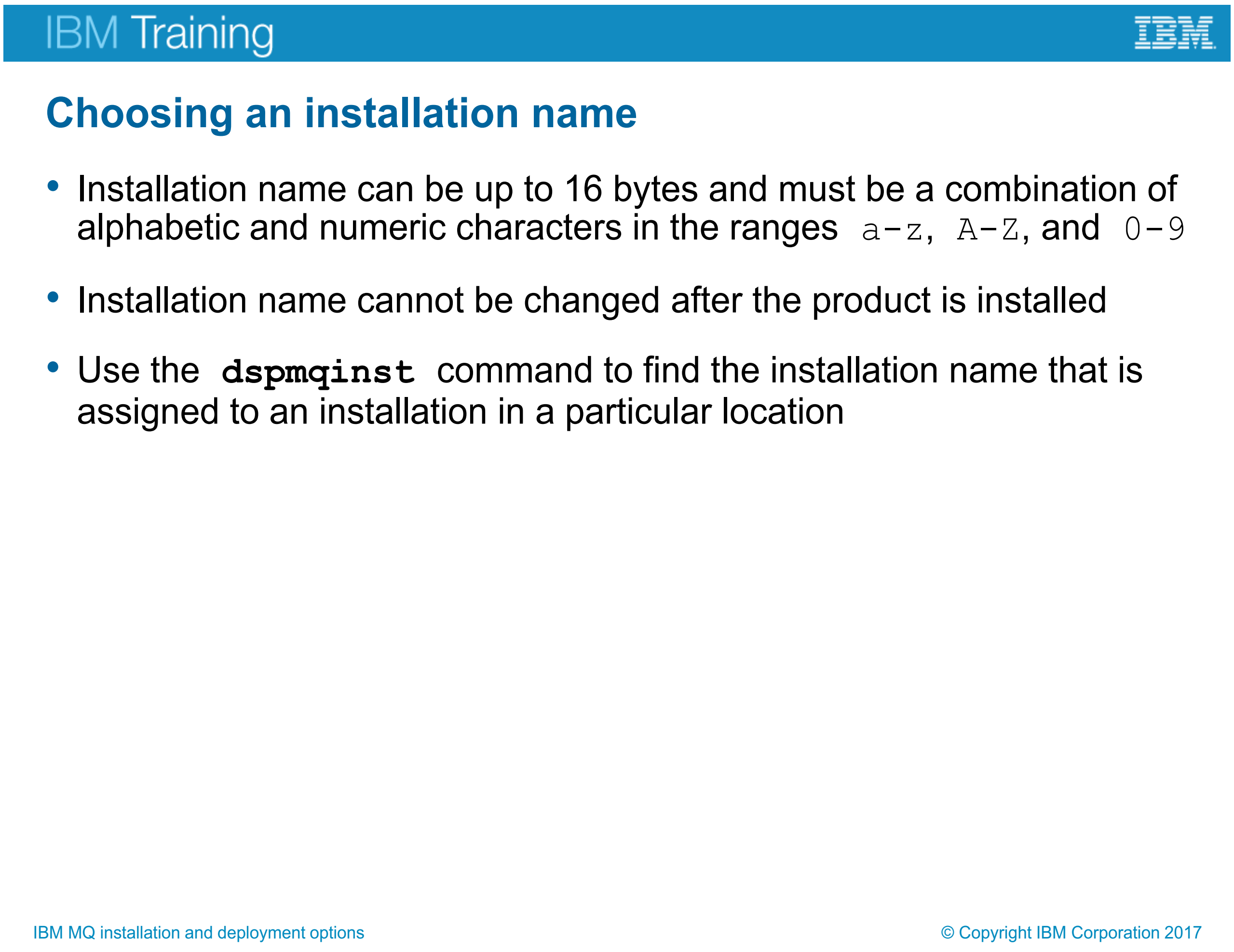
This figure shows some examples of the commands that assist with installation management.

The **dspmq** and **dspmqver** commands show all of the installations, their directories, and the associated queue managers.

The queue manager control commands must be run from the correct directory. If you try to run them against a queue manager that is not associated with that installation, they return an error.

Optionally, you can also use the **setmqenv** command to automatically set up the environment for a different installation of MQ.

Multiple instance management is described in detail in the WM213/ZM213, *IBM MQ V9 Advanced System Administration*.



IBM Training

Choosing an installation name

- Installation name can be up to 16 bytes and must be a combination of alphabetic and numeric characters in the ranges `a-z`, `A-Z`, and `0-9`
- Installation name cannot be changed after the product is installed
- Use the **`dspmqinst`** command to find the installation name that is assigned to an installation in a particular location

IBM MQ installation and deployment options © Copyright IBM Corporation 2017

Figure 2-16. Choosing an installation name

Each installation of IBM MQ on UNIX, Linux, and Windows has a unique identifier that is known as an *installation name*. The installation name associates objects such as queue managers and configuration files with an installation.

You can choose an installation name that is meaningful to you. For example, you might call a test system ***testMQ***.

An installation name can be up to 16 bytes and must be a combination of alphabetic and numeric characters in the ranges a – z, A – Z, and 0 – 9. You cannot use blank characters. The installation name must be unique, regardless of whether uppercase or lowercase characters are used. For example, the names ***INSTALLATIONNAME*** and ***InstallationName*** are not unique.

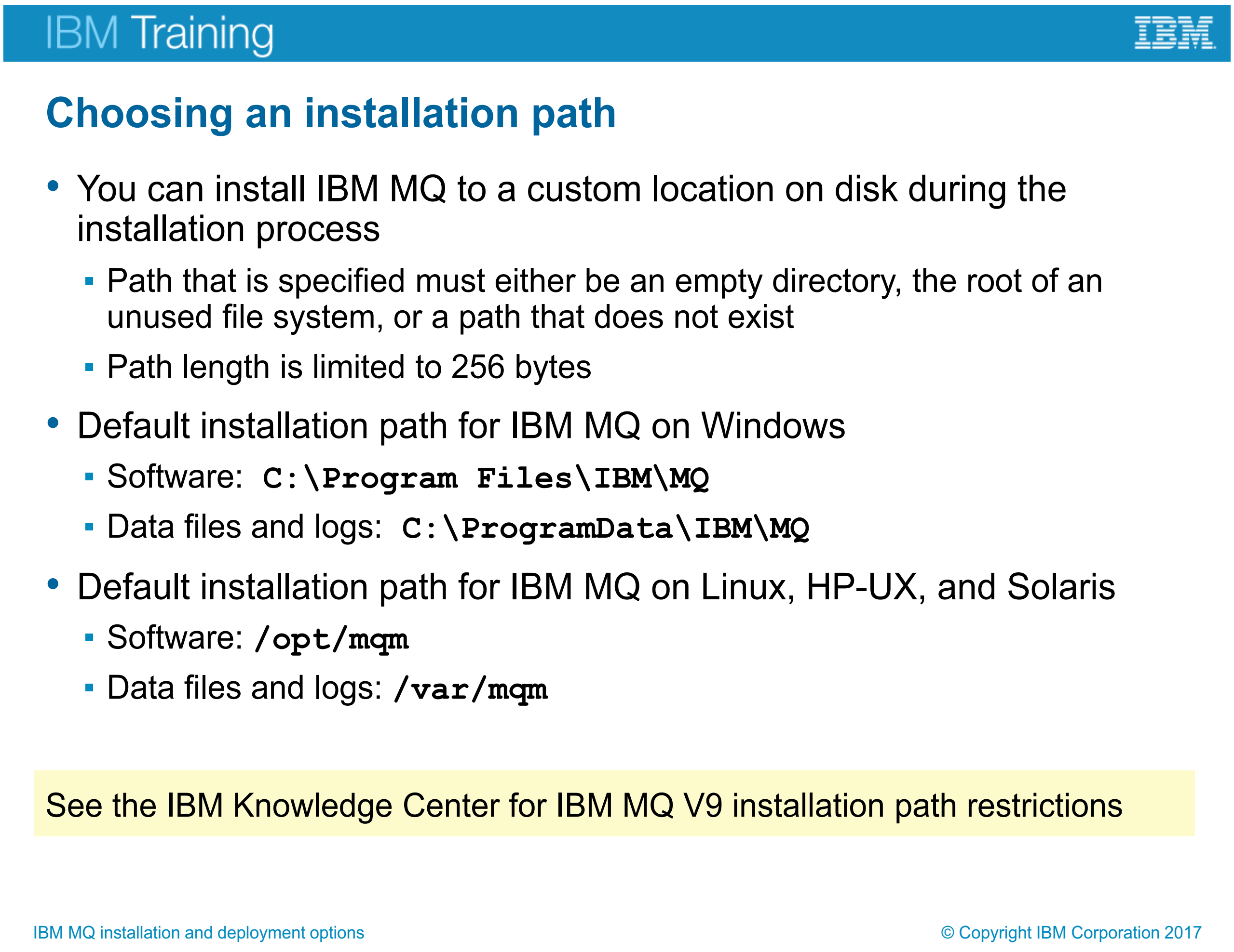
If you do not specify an installation name when MQ is installed, a default installation name is automatically assigned. For the first installation, this name is ***Installation1***. For the second installation, the name is ***Installation2***. The installation name ***Installation0*** is reserved for an installation of WebSphere MQ Version 7.0.1.

On UNIX and Linux, the first IBM MQ installation is automatically given an installation name of ***Installation1***. For subsequent installations, you can use the **`crtmqinst`** command to set the installation name before you install MQ.

On Windows, you can choose the installation name during the installation process.

The installation name cannot be changed after MQ is installed.

You can find out what installation name is assigned to an installation in a particular location by using the **`dspmqinst`** command.



IBM Training

Choosing an installation path

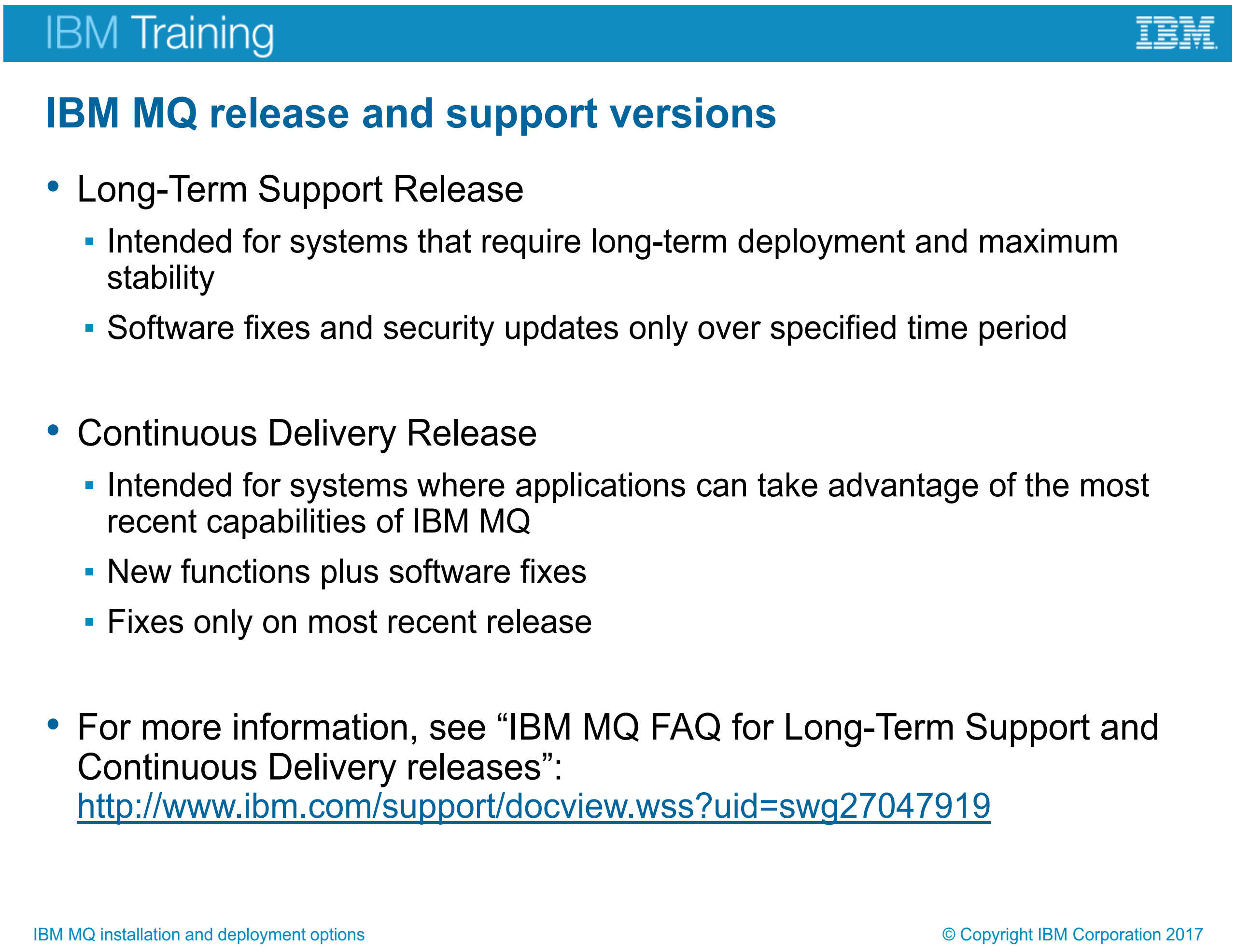
- You can install IBM MQ to a custom location on disk during the installation process
  - Path that is specified must either be an empty directory, the root of an unused file system, or a path that does not exist
  - Path length is limited to 256 bytes
- Default installation path for IBM MQ on Windows
  - Software: **`C:\Program Files\IBM\MQ`**
  - Data files and logs: **`C:\ProgramData\IBM\MQ`**
- Default installation path for IBM MQ on Linux, HP-UX, and Solaris
  - Software: **`/opt/mqm`**
  - Data files and logs: **`/var/mqm`**

See the IBM Knowledge Center for IBM MQ V9 installation path restrictions

IBM MQ installation and deployment options © Copyright IBM Corporation 2017

Figure 2-17. Choosing an installation path

You can install MQ to a default or customer directory during the installation process. If you specify a custom path, it must be either an empty directory, the root of an unused file system, or a path to a directory that does not exist.



IBM Training

IBM MQ release and support versions

- Long-Term Support Release
  - Intended for systems that require long-term deployment and maximum stability
  - Software fixes and security updates only over specified time period
- Continuous Delivery Release
  - Intended for systems where applications can take advantage of the most recent capabilities of IBM MQ
  - New functions plus software fixes
  - Fixes only on most recent release
- For more information, see “IBM MQ FAQ for Long-Term Support and Continuous Delivery releases”: http://www.ibm.com/support/docview.wss?uid=swg27047919

IBM MQ installation and deployment options © Copyright IBM Corporation 2017

Figure 2-18. IBM MQ release and support versions

The next step in an on-premises installation is to obtain the base code and fix packs. To accelerate the speed of application development, IBM MQ Version 9.0 introduces a new continuous delivery and support mode for MQ base code and fix packs.

This new delivery model allows developers to access the most recent product enhancements and administrators to update their MQ environments with software fixes only.

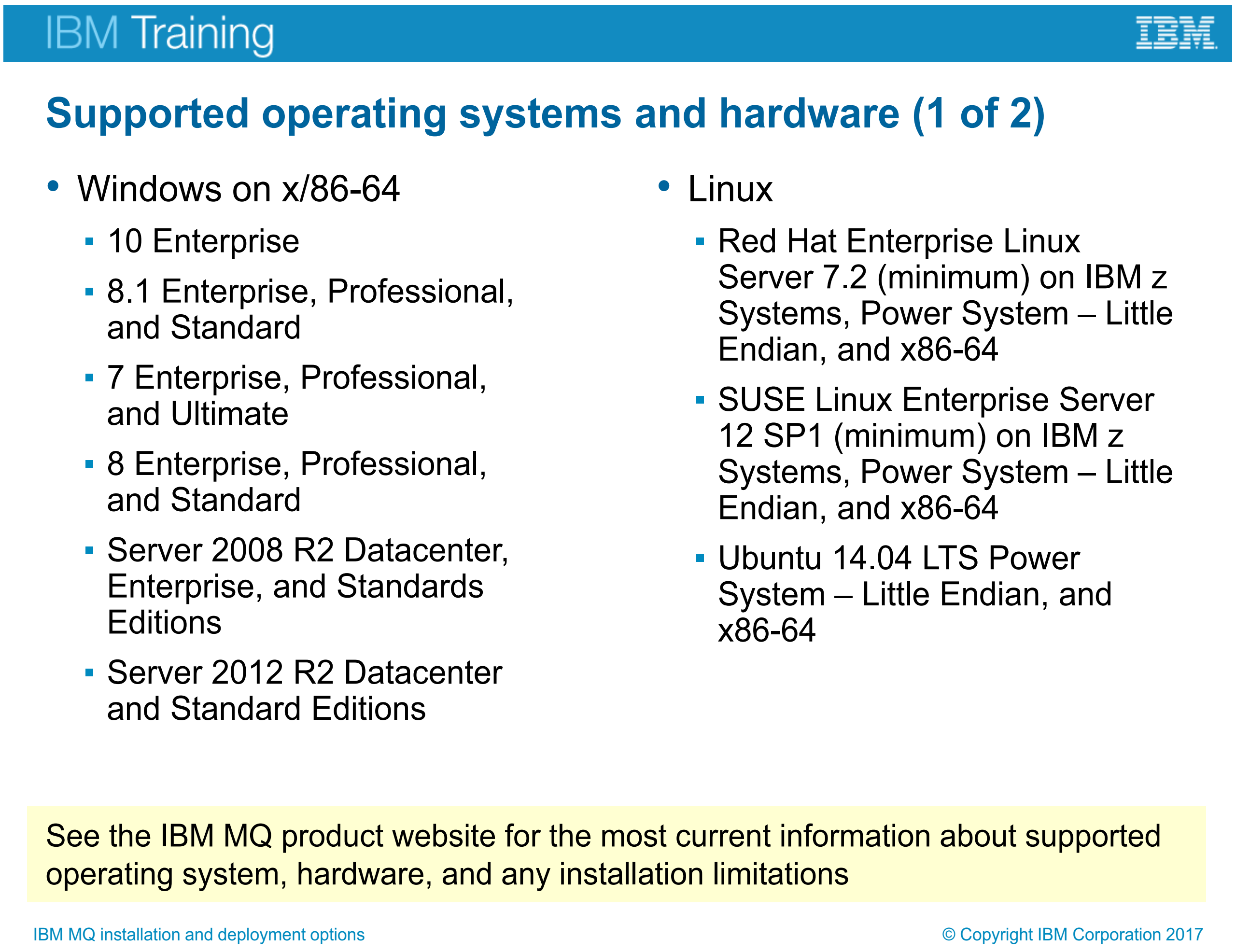
This new delivery model also allows IBM to respond faster to enhancement requests and opportunities for expanding MQ availability on new infrastructures and environments.

The two MQ release versions for IBM MQ Version 9.0 are the Long Term Support Release (LTSR) and Continuous Delivery Release (CDR).

The Long-Term Support Release version is the product level for which support, including defect and security updates, is provided over a specified time. This version is intended for systems that require maximum stability.

The Continuous Delivery Release version delivers new functional enhancements, and fixes and security updates, on a much shorter release cycle. This version provides rapid access to new functions. It is intended for systems where enterprises want to use the newest capabilities of MQ.

For more information about the MQ release versions, see the “IBM MQ FAQ for Long-Term Support and Continuous releases” web page at:
http://www.ibm.com/support/docview.wss?uid=swg27047919



IBM Training

Supported operating systems and hardware (1 of 2)

- Windows on x/86-64
  - 10 Enterprise
  - 8.1 Enterprise, Professional, and Standard
  - 7 Enterprise, Professional, and Ultimate
  - 8 Enterprise, Professional, and Standard
  - Server 2008 R2 Datacenter, Enterprise, and Standards Editions
  - Server 2012 R2 Datacenter and Standard Editions
- Linux
  - Red Hat Enterprise Linux Server 7.2 (minimum) on IBM z Systems, Power System – Little Endian, and x86-64
  - SUSE Linux Enterprise Server 12 SP1 (minimum) on IBM z Systems, Power System – Little Endian, and x86-64
  - Ubuntu 14.04 LTS Power System – Little Endian, and x86-64

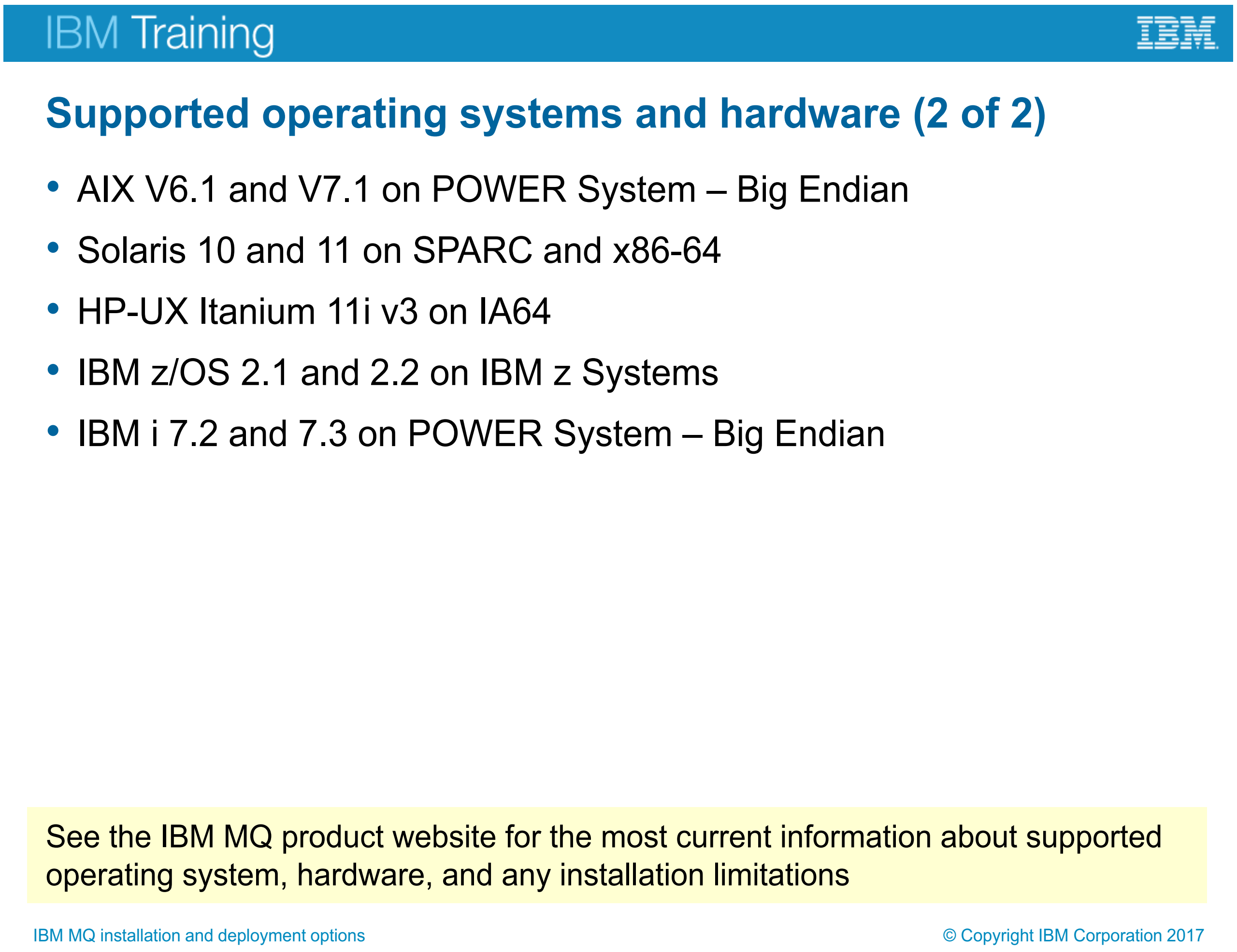
See the IBM MQ product website for the most current information about supported operating system, hardware, and any installation limitations

IBM MQ installation and deployment options © Copyright IBM Corporation 2017

Figure 2-19. Supported operating systems and hardware (1 of 2)

This figure lists the supported operating systems for IBM MQ server and client installation on Windows and Linux.

For details of the specific operating system and hardware requirements, see the IBM MQ product documentation.



IBM Training

Supported operating systems and hardware (2 of 2)

- AIX V6.1 and V7.1 on POWER System – Big Endian
- Solaris 10 and 11 on SPARC and x86-64
- HP-UX Itanium 11i v3 on IA64
- IBM z/OS 2.1 and 2.2 on IBM z Systems
- IBM i 7.2 and 7.3 on POWER System – Big Endian

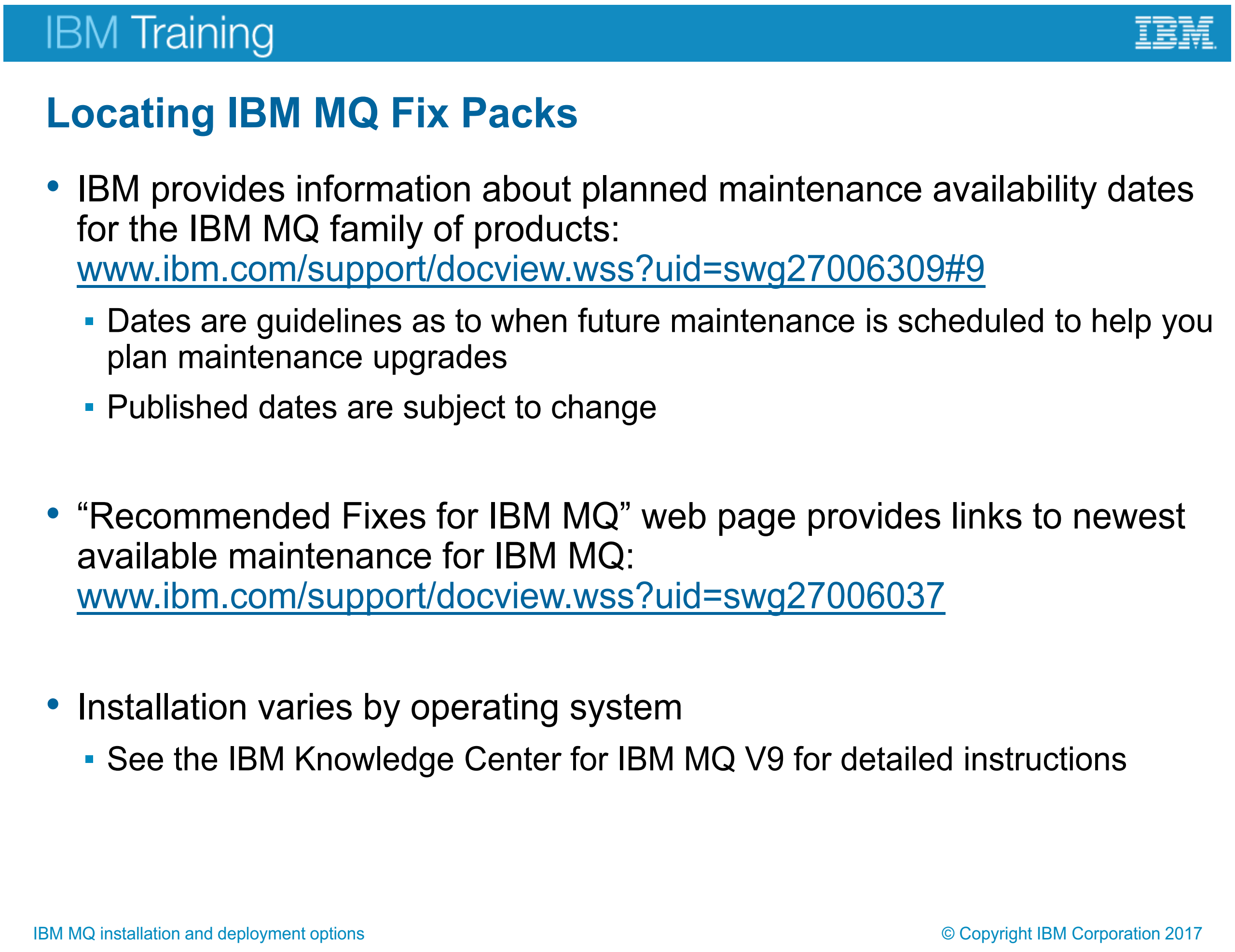
See the IBM MQ product website for the most current information about supported operating system, hardware, and any installation limitations

IBM MQ installation and deployment options © Copyright IBM Corporation 2017

Figure 2-20. Supported operating systems and hardware (2 of 2)

This figure lists the supported operating systems for IBM MQ server and client installation on AIX, Solaris, HP-UX, IBM z/OS, and IBM i.

Always check the IBM MQ product documentation and the IBM MQ product website for the most current and detailed information about supported operating systems.



IBM Training

IBM

Locating IBM MQ Fix Packs

- IBM provides information about planned maintenance availability dates for the IBM MQ family of products: www.ibm.com/support/docview.wss?uid=swg27006309#9
  - Dates are guidelines as to when future maintenance is scheduled to help you plan maintenance upgrades
  - Published dates are subject to change
- “Recommended Fixes for IBM MQ” web page provides links to newest available maintenance for IBM MQ: www.ibm.com/support/docview.wss?uid=swg27006037
- Installation varies by operating system
  - See the IBM Knowledge Center for IBM MQ V9 for detailed instructions

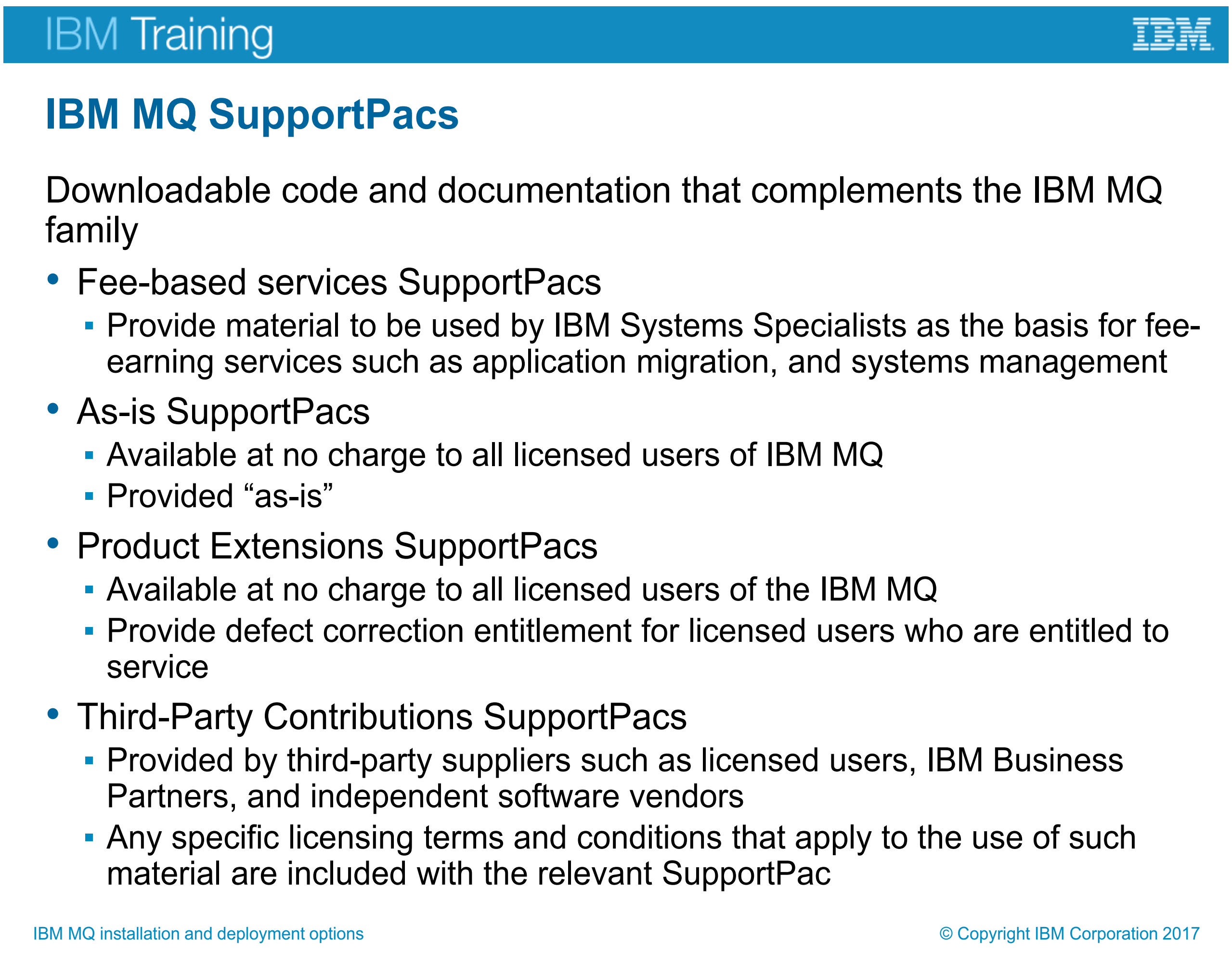
IBM MQ installation and deployment options

© Copyright IBM Corporation 2017

Figure 2-21. Locating IBM MQ Fix Packs

IBM provides software updates to MQ in the form of fix packs. Before you install MQ, determine whether any fix packs are available for the version of MQ that you planning to install.

This figure identifies websites that contain information about IBM MQ Fix Packs.



IBM Training

IBM MQ SupportPacs

Downloadable code and documentation that complements the IBM MQ family

- Fee-based services SupportPacs
  - Provide material to be used by IBM Systems Specialists as the basis for fee-earning services such as application migration, and systems management
- As-is SupportPacs
  - Available at no charge to all licensed users of IBM MQ
  - Provided “as-is”
- Product Extensions SupportPacs
  - Available at no charge to all licensed users of the IBM MQ
  - Provide defect correction entitlement for licensed users who are entitled to service
- Third-Party Contributions SupportPacs
  - Provided by third-party suppliers such as licensed users, IBM Business Partners, and independent software vendors
  - Any specific licensing terms and conditions that apply to the use of such material are included with the relevant SupportPac

IBM MQ installation and deployment options © Copyright IBM Corporation 2017

Figure 2-22. IBM MQ SupportPacs

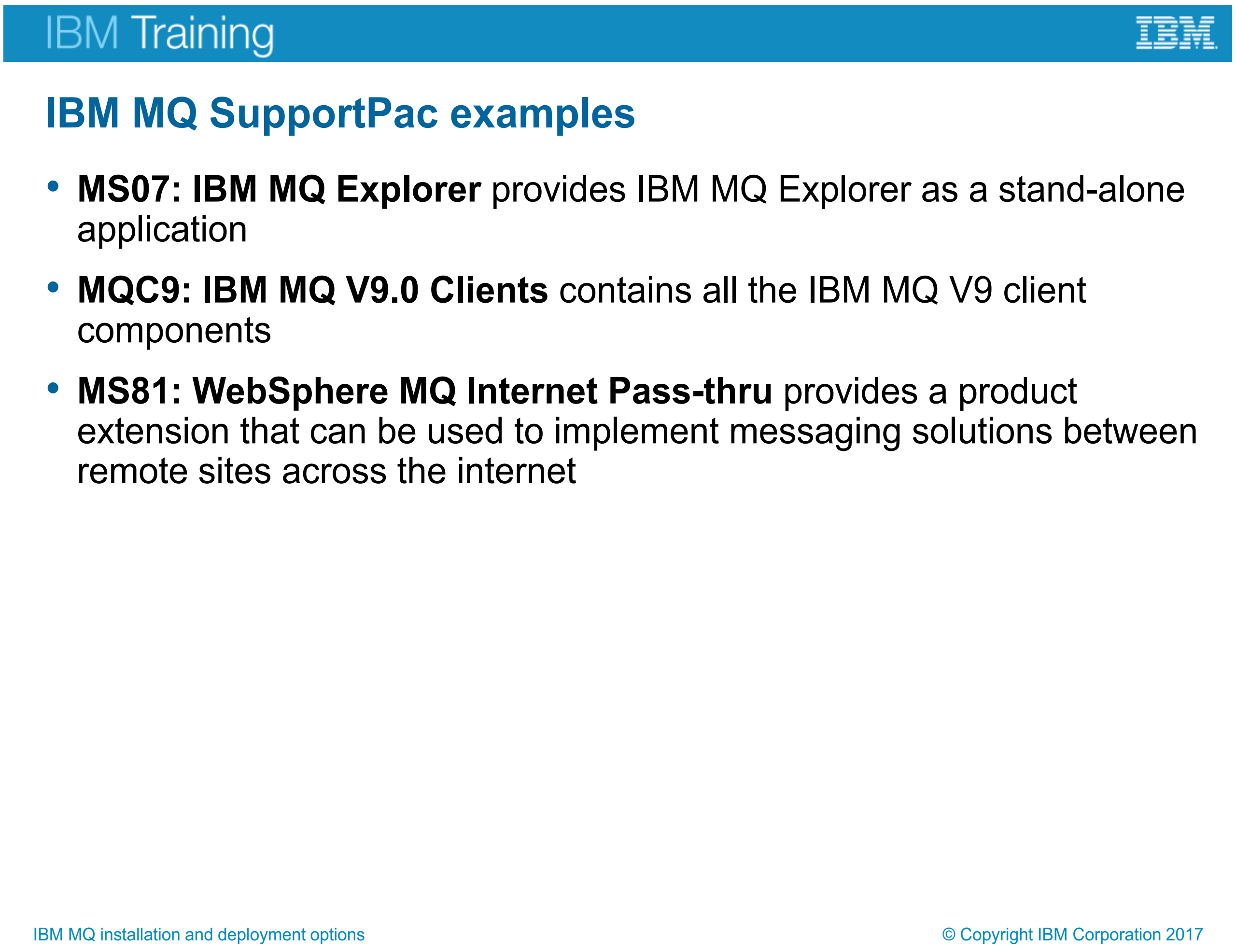
The third step in an MQ on-premises installation is to evaluate the MQ SupportPacs.

MQ SupportPacs provide you with a wide range of downloadable code and documentation that complements the MQ family of products.

SupportPacs are available in four categories:

- Fee-based services (not available for download).
- Freeware that is provided “as is” with no warranty or defect correction.
- Fully supported product extensions that IBM develops. SupportPacs in this category are available at no charge to all licensed users of MQ products on the operating systems that are specified in the SupportPac description. They are supplied under the standard terms and conditions of the IBM International Program License Agreement (IPLA) and of the associated license information. They provide a defect correction entitlement for those licensed users who are entitled to service for IBM MQ. IBM reserves the right to discontinue service on the SupportPac when it is withdrawn from marketing by IBM.
- Vendor contributions that are provided “as is” with no warranty or defect correction. SupportPacs in this category are provided by third-party suppliers such as licensed users, and IBM Business Partners. While these SupportPacs are downloadable from the IBM website, any

specific licensing terms and conditions that apply to the use of the material are included with the relevant SupportPac.



IBM Training

IBM MQ SupportPac examples

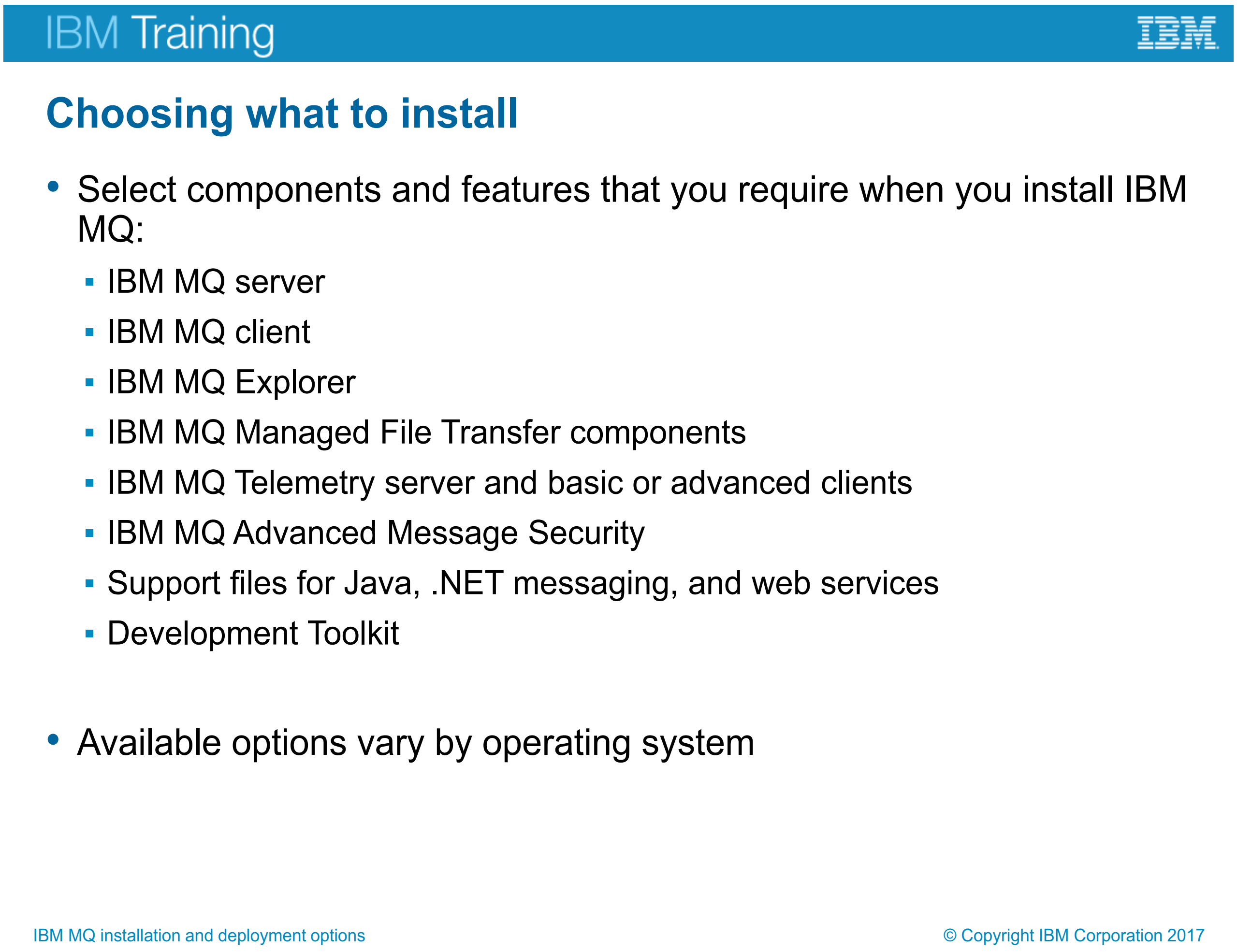
- **MS07: IBM MQ Explorer** provides IBM MQ Explorer as a stand-alone application
- **MQC9: IBM MQ V9.0 Clients** contains all the IBM MQ V9 client components
- **MS81: WebSphere MQ Internet Pass-thru** provides a product extension that can be used to implement messaging solutions between remote sites across the internet

IBM MQ installation and deployment options

© Copyright IBM Corporation 2017

Figure 2-23. IBM MQ SupportPac examples

This figure lists some examples of IBM MQ SupportPacs.



IBM Training

Choosing what to install

- Select components and features that you require when you install IBM MQ:
  - IBM MQ server
  - IBM MQ client
  - IBM MQ Explorer
  - IBM MQ Managed File Transfer components
  - IBM MQ Telemetry server and basic or advanced clients
  - IBM MQ Advanced Message Security
  - Support files for Java, .NET messaging, and web services
  - Development Toolkit
- Available options vary by operating system

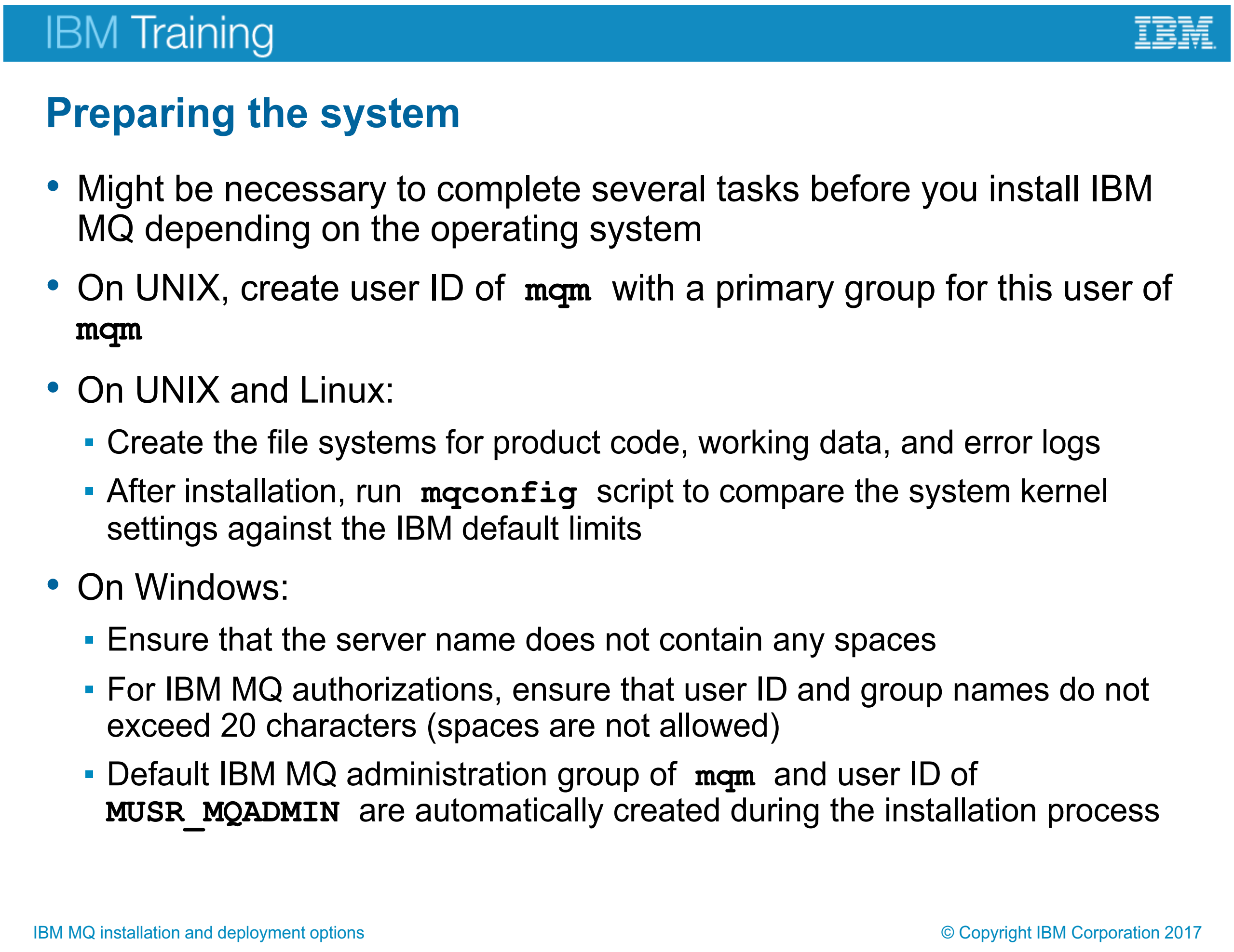
IBM MQ installation and deployment options © Copyright IBM Corporation 2017

Figure 2-24. Choosing what to install

The next step in an MQ on-premises installation is to choose which MQ components you are going to install. Is this MQ installation for a developer or for production? Does this installation require IBM MQ Managed File Transfer or an IBM MQ Telemetry server?

As noted on the figure, the available options and components vary by operating system.

MQ has a typical, a compact, and a custom installation option. To ensure that the installation completes, select **Custom** so that you can select all the features. If you choose a **Typical** installation, you can easily miss new features.



IBM Training

Preparing the system

- Might be necessary to complete several tasks before you install IBM MQ depending on the operating system
- On UNIX, create user ID of **`mqm`** with a primary group for this user of **`mqm`**
- On UNIX and Linux:
  - Create the file systems for product code, working data, and error logs
  - After installation, run **`mqconfig`** script to compare the system kernel settings against the IBM default limits
- On Windows:
  - Ensure that the server name does not contain any spaces
  - For IBM MQ authorizations, ensure that user ID and group names do not exceed 20 characters (spaces are not allowed)
  - Default IBM MQ administration group of **`mqm`** and user ID of **`MUSR_MQADMIN`** are automatically created during the installation process

IBM MQ installation and deployment options © Copyright IBM Corporation 2017

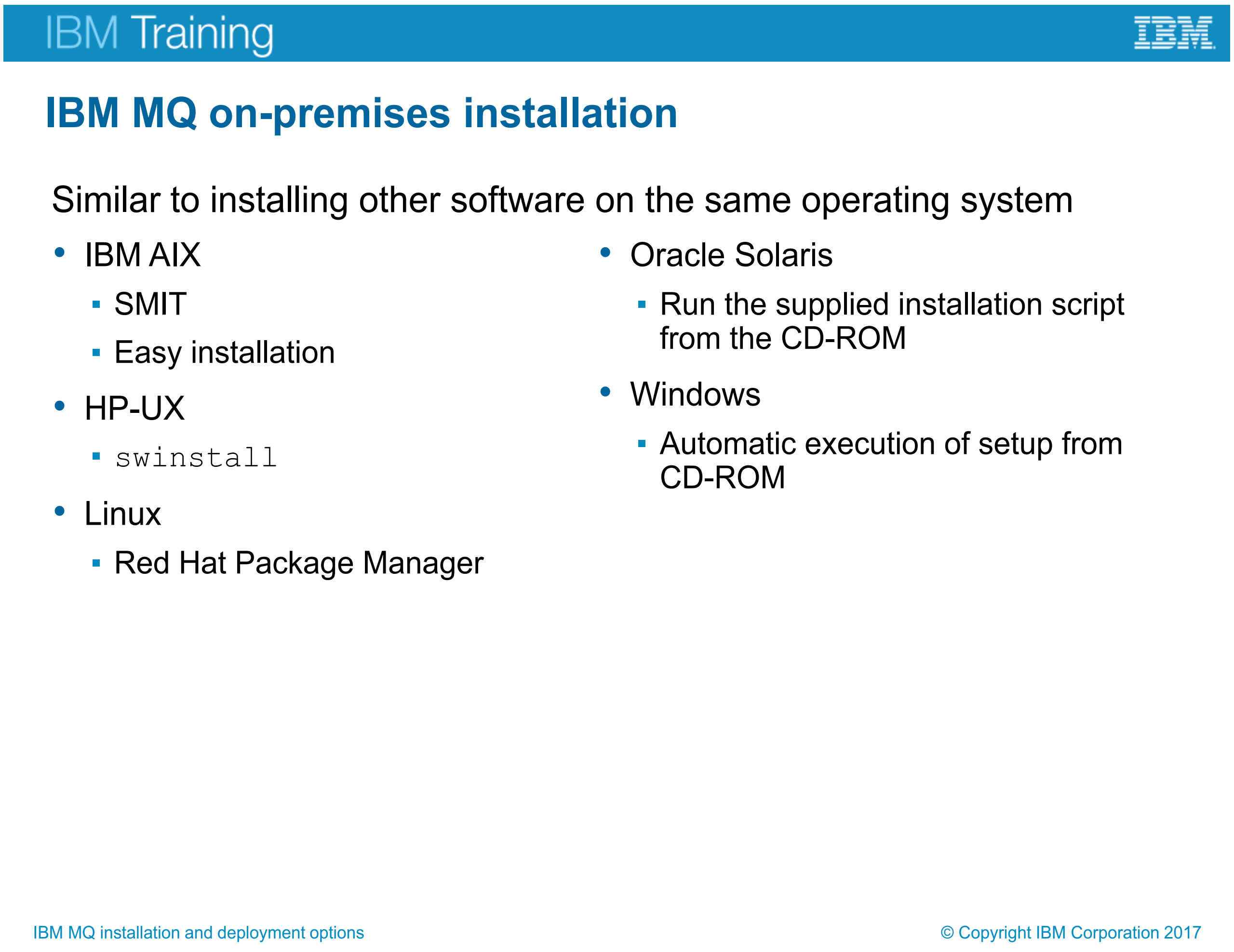
Figure 2-25. Preparing the system

The next step in an MQ on-premises installation is to prepare the system.

On UNIX, you must create a user ID and group that is named **`mqm`**. In most cases, the Linux installation program automatically creates this user and group.

On UNIX and Linux, run the **`mqconfig`** script to compare the system kernel settings against the IBM default limits, which should be sufficient for many systems. However, these settings might be insufficient if you have a busy system with high production volumes, or if your system is also hosting other products like databases or application servers. Run the **`mqconfig`** script when your system is processing its peak volume to ensure that the actual resource usage, which is shown as a percentage, is not excessively high. The **`mqconfig`** script is described in the IBM technote "How to configure UNIX and Linux systems for IBM MQ" at http://www.ibm.com/support/docview.wss?uid=swg21271236

On Windows, ensure that the name of the server does not contain any spaces. Also, ensure that any user ID and group names do not exceed 20 characters. The MQ administrator group and user ID are automatically created during the installation process.



IBM Training

IBM MQ on-premises installation

Similar to installing other software on the same operating system

- IBM AIX
  - SMIT
  - Easy installation
- HP-UX
  - `swinstall`
- Linux
  - Red Hat Package Manager
- Oracle Solaris
  - Run the supplied installation script from the CD-ROM
- Windows
  - Automatic execution of setup from CD-ROM

IBM MQ installation and deployment options

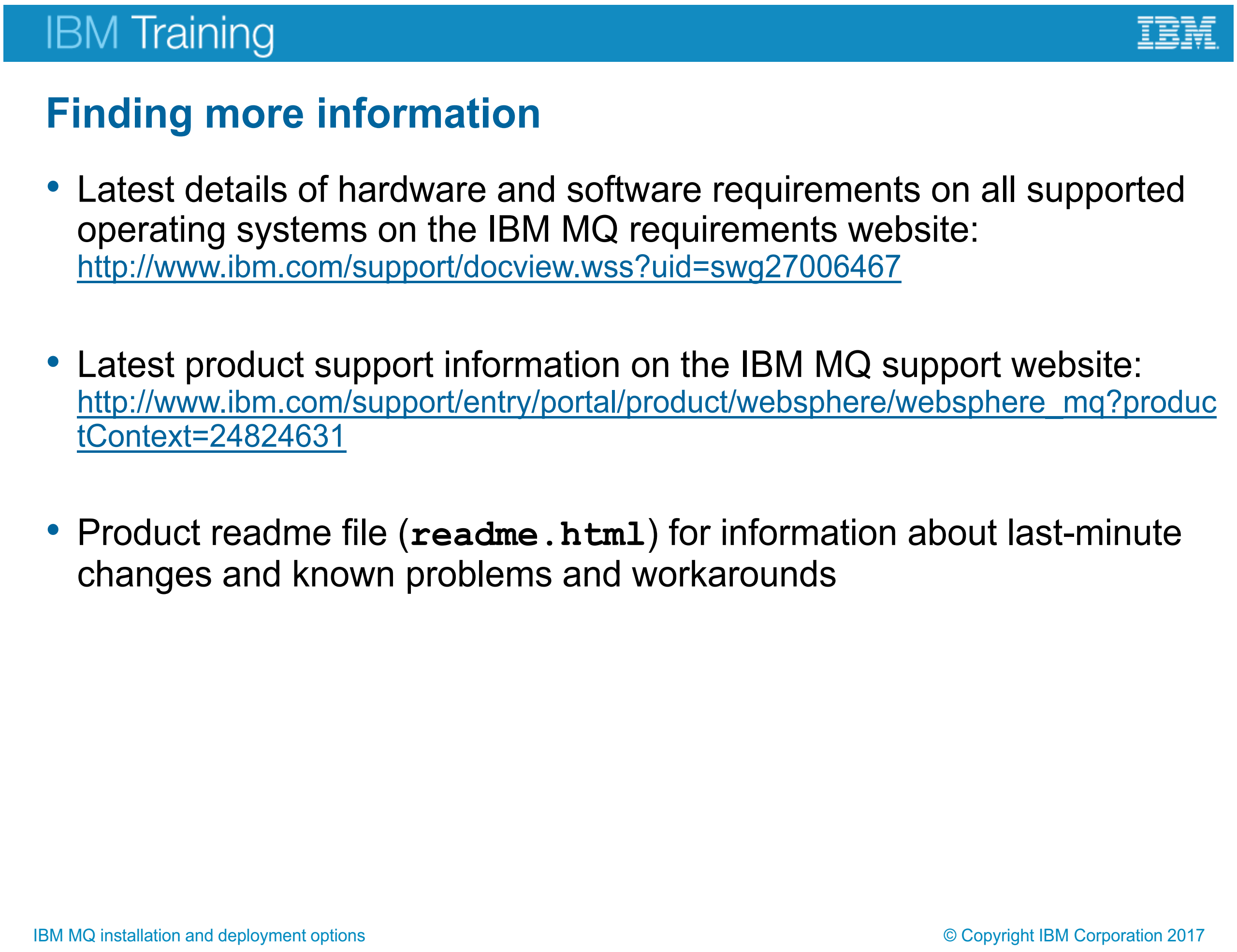
© Copyright IBM Corporation 2017

Figure 2-26. IBM MQ on-premises installation

Installing IBM MQ is like installing any other software on the same operating system and uses many of the same installation tools.

For example, you can use SMIT to install MQ for AIX, or you can choose the easy installation. The easy installation places a minimal, typical set of components on your system. It excludes, for example, the online documentation and the application development support.

During installation, MQ automatically creates an MQ configuration file, which contains information relevant to all queue managers that are created on the system. Only one MQ configuration file exists for each system. You learn more about this configuration file later in this course.



IBM Training

Finding more information

- Latest details of hardware and software requirements on all supported operating systems on the IBM MQ requirements website: http://www.ibm.com/support/docview.wss?uid=swg27006467
- Latest product support information on the IBM MQ support website: http://www.ibm.com/support/entry/portal/product/websphere/websphere_mq?productContext=24824631
- Product readme file (**`readme.html`**) for information about last-minute changes and known problems and workarounds

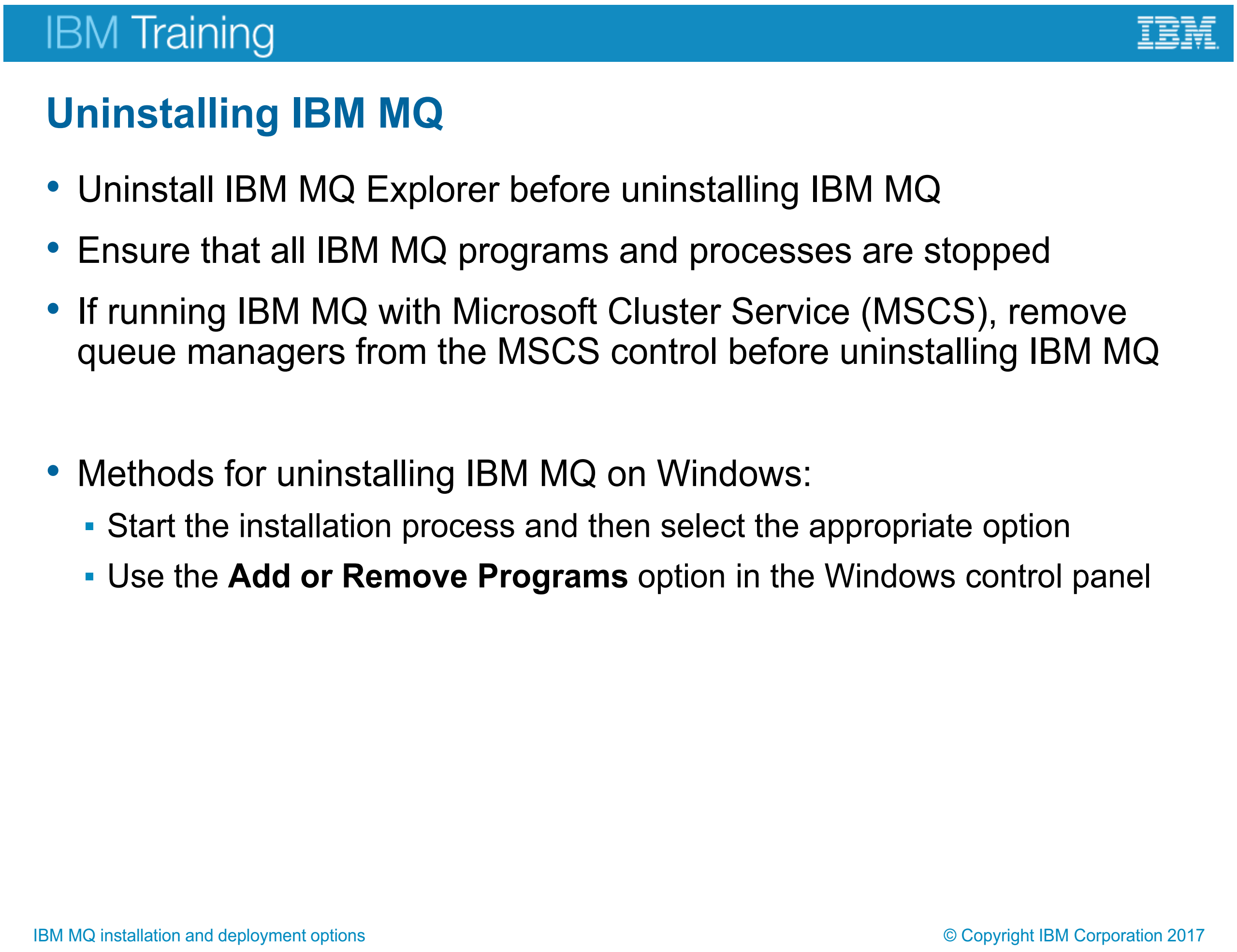
IBM MQ installation and deployment options

© Copyright IBM Corporation 2017

Figure 2-27. Finding more information

Always review the IBM MQ product websites to see the most recent specifications, requirements, and product support information.

The MQ installation files include a readme file that contains information about last-minute changes, known problems, and workarounds.



IBM Training

Uninstalling IBM MQ

- Uninstall IBM MQ Explorer before uninstalling IBM MQ
- Ensure that all IBM MQ programs and processes are stopped
- If running IBM MQ with Microsoft Cluster Service (MSCS), remove queue managers from the MSCS control before uninstalling IBM MQ

- Methods for uninstalling IBM MQ on Windows:
  - Start the installation process and then select the appropriate option
  - Use the **Add or Remove Programs** option in the Windows control panel

IBM MQ installation and deployment options © Copyright IBM Corporation 2017

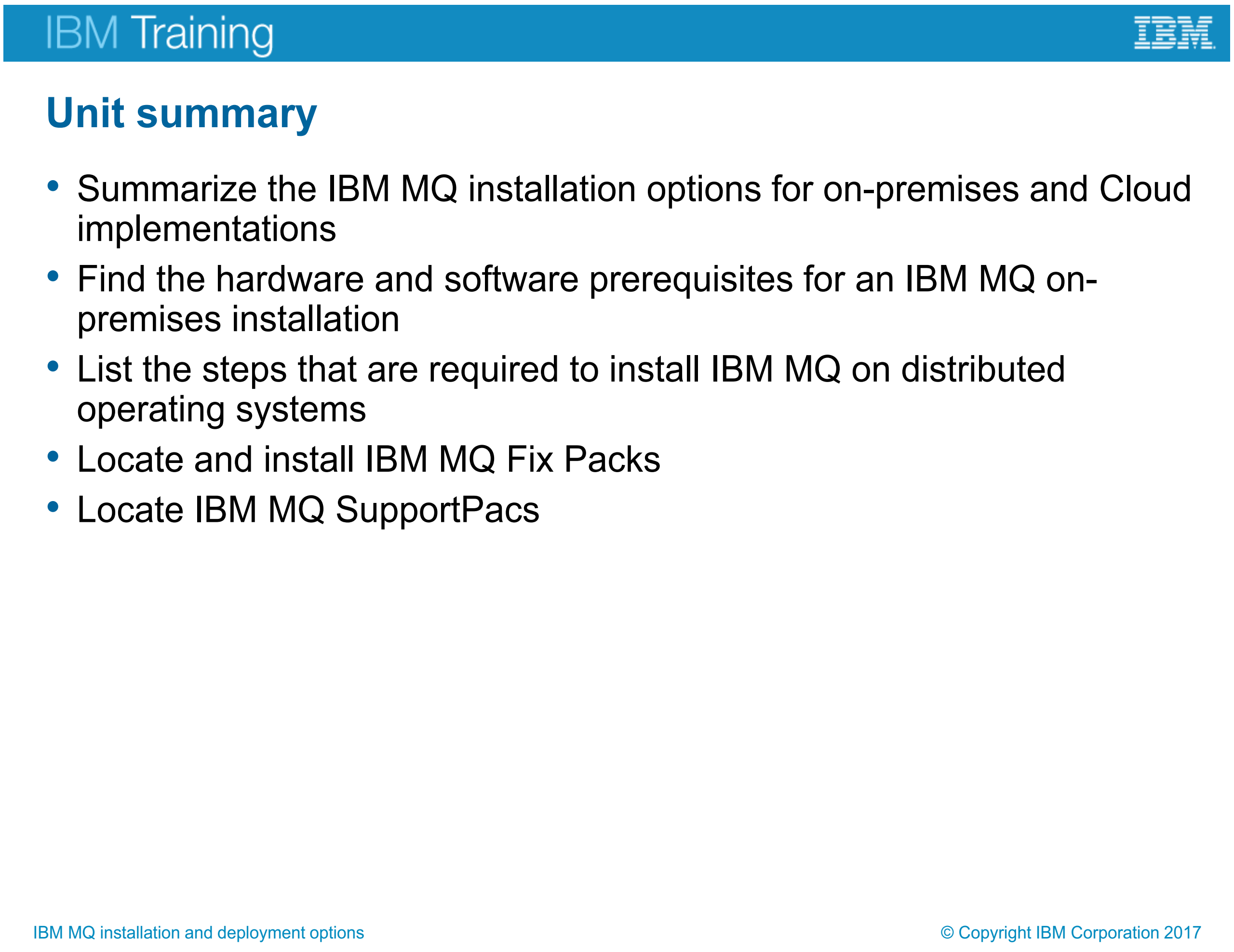
Figure 2-28. Uninstalling IBM MQ

If necessary, you can uninstall MQ. Ensure that the applications and services are stopped before you attempt to uninstall IBM MQ.

On Windows, you can use the **Add or Remove Programs** application in the Windows **Control Panel** to uninstall MQ components and services.

You can also uninstall MQ from a command if it was installed in the default location. For example, on Linux, you would complete the following steps to uninstall MQ:

1. Type: **`installp -u mqm`**
2. Delete the contents of the MQ data directory.



IBM Training

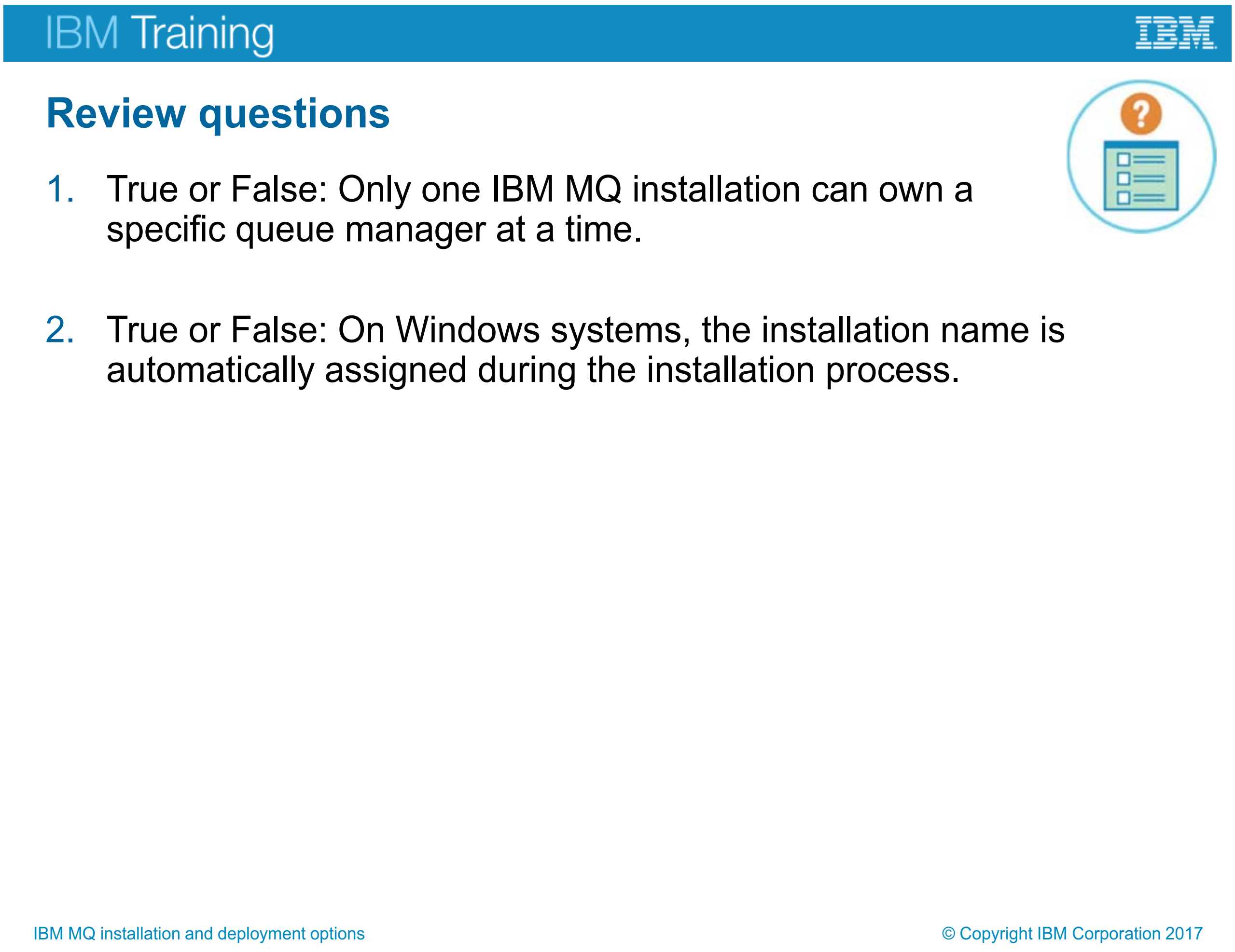
Unit summary

- Summarize the IBM MQ installation options for on-premises and Cloud implementations
- Find the hardware and software prerequisites for an IBM MQ on-premises installation
- List the steps that are required to install IBM MQ on distributed operating systems
- Locate and install IBM MQ Fix Packs
- Locate IBM MQ SupportPacs

IBM MQ installation and deployment options

© Copyright IBM Corporation 2017

Figure 2-29. Unit summary



IBM Training

Review questions

1. True or False: Only one IBM MQ installation can own a specific queue manager at a time.

2. True or False: On Windows systems, the installation name is automatically assigned during the installation process.

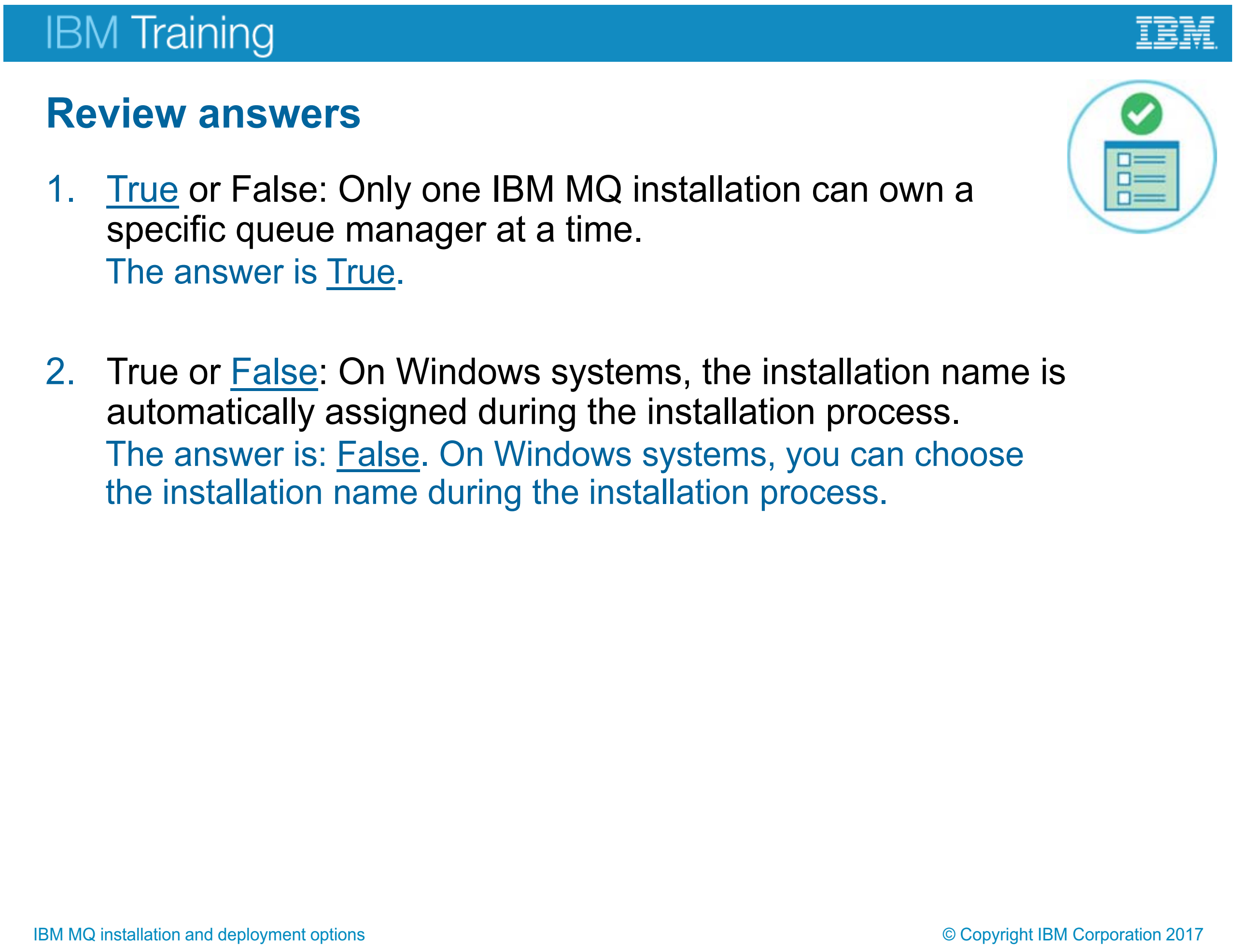
IBM MQ installation and deployment options © Copyright IBM Corporation 2017

Figure 2-30. Review questions

Write your answers here:

1.

2.



IBM Training

Review answers

1. True or False: Only one IBM MQ installation can own a specific queue manager at a time.
   The answer is True.

2. True or False: On Windows systems, the installation name is automatically assigned during the installation process.
   The answer is: False. On Windows systems, you can choose the installation name during the installation process.

IBM MQ installation and deployment options

© Copyright IBM Corporation 2017

Figure 2-31. Review answers

Unit 3. Creating a queue manager and queues

Estimated time

01:00

Overview

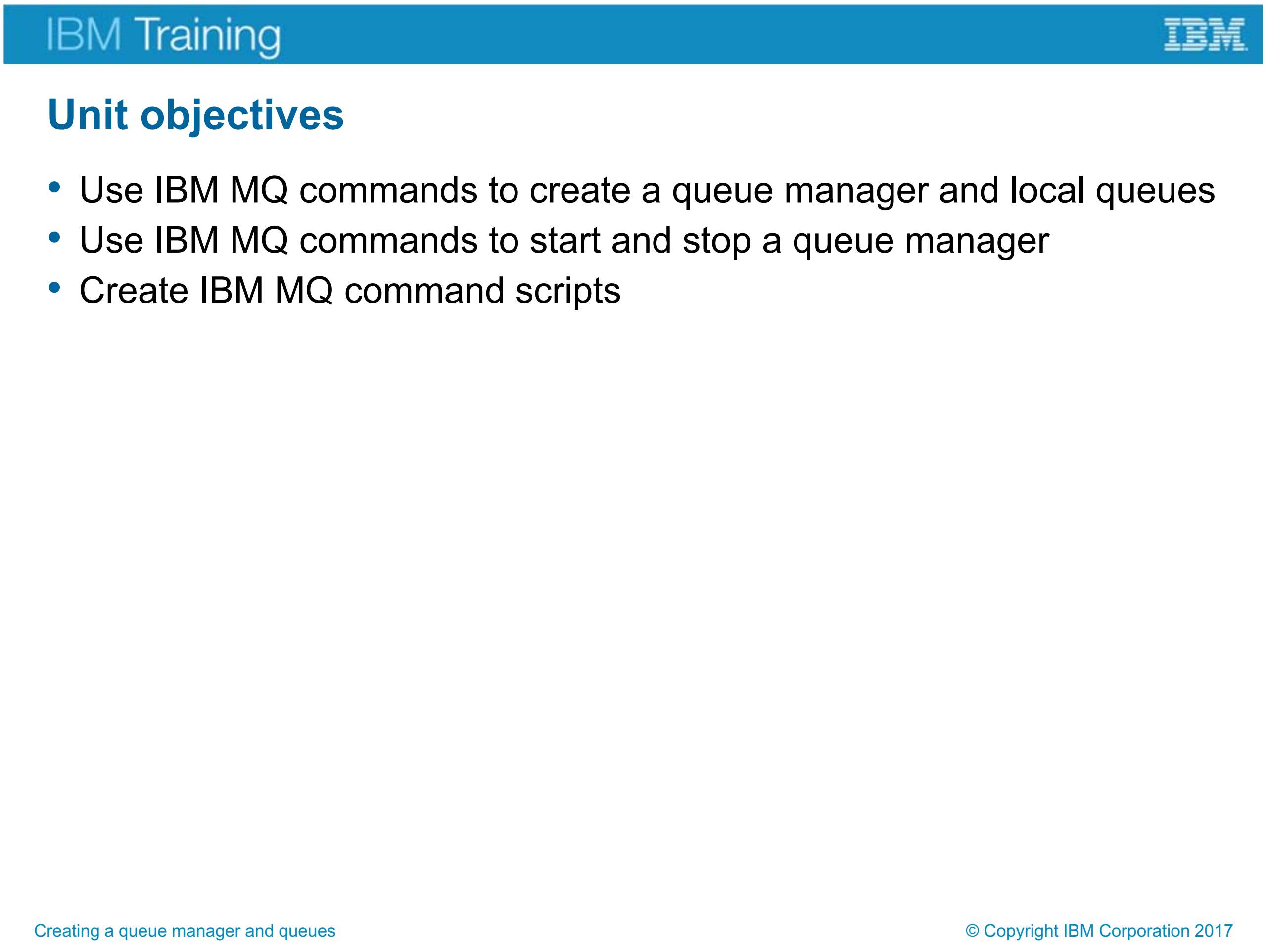
In this unit, you learn how to use the IBM MQ commands and command scripts to verify an installation and create a queue manager and local queues.

How you will check your progress

- Review questions
- Hands-on exercise

References

IBM Knowledge Center for IBM MQ V9



IBM Training

Unit objectives

- Use IBM MQ commands to create a queue manager and local queues
- Use IBM MQ commands to start and stop a queue manager
- Create IBM MQ command scripts

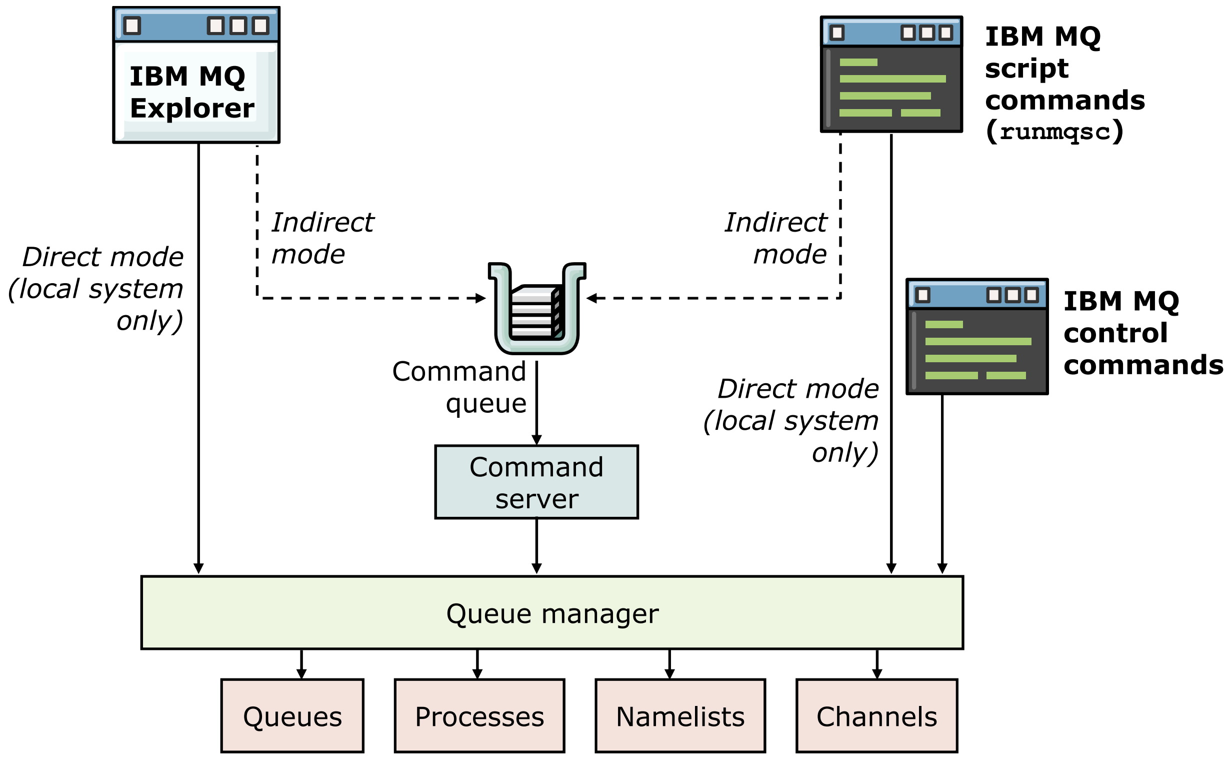
Creating a queue manager and queues

© Copyright IBM Corporation 2017

Figure 3-1. Unit objectives

IBM Training

Administration interfaces



Creating a queue manager and queues

© Copyright IBM Corporation 2017

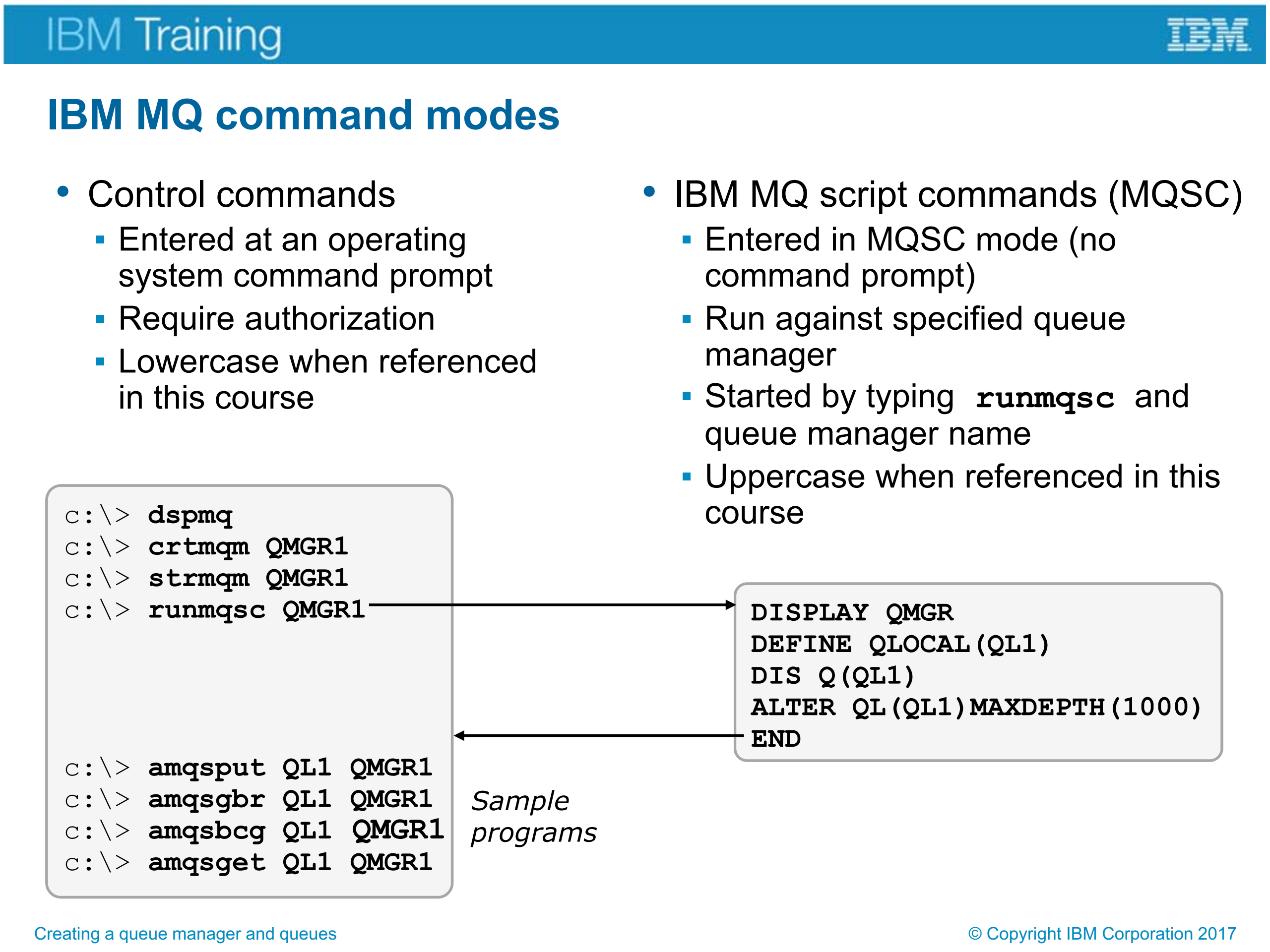
Figure 3-2. Administration interfaces

The primary IBM MQ administration interfaces for MQ on distributed operating systems are:

- MQ Explorer
- MQ script commands
- MQ control commands

MQ Explorer is a graphical user interface that runs on Linux and Windows. It can connect to, control, and configure MQ on all operating systems. IBM MQ Explorer is described in detail later in this course.

MQ script commands and MQ control commands are described in more detail in this unit and throughout this course.



IBM Training

IBM MQ command modes

- Control commands
  - Entered at an operating system command prompt
  - Require authorization
  - Lowercase when referenced in this course
- IBM MQ script commands (MQSC)
  - Entered in MQSC mode (no command prompt)
  - Run against specified queue manager
  - Started by typing **runmqsc** and queue manager name
  - Uppercase when referenced in this course

```
c:\> dspmq
c:\> crtmqm QMGR1
c:\> strmqm QMGR1
c:\> runmqsc QMGR1
```

```
DISPLAY QMGR
DEFINE QLOCAL(QL1)
DIS Q(QL1)
ALTER QL(QL1)MAXDEPTH(1000)
END
```

```
c:\> amqsput QL1 QMGR1
c:\> amqsgbr QL1 QMGR1
c:\> amqsbcg QL1 QMGR1
c:\> amqsget QL1 QMGR1
```

Sample programs

Creating a queue manager and queues © Copyright IBM Corporation 2017

Figure 3-3. IBM MQ command modes

As shown in the figure, MQ supports two command modes: control command mode and MQSC mode.

Control commands are entered at the operating system command prompt. They can also be included in an operating system command file such as a shell script on UNIX or a batch file in Windows. The sample programs that are included with MQ are run with control commands.

You need the following authority to use a control command.

- On UNIX and Linux, your user ID must belong to the group “mqm”, but not necessarily as its primary group.
- On Windows, your user ID must belong either to the “mqm” local group or to the “Administrators” local group. It is possible for your user ID to belong to a global group, which, in turn, belongs to either of these local groups.

MQSC mode is started by entering the **runmqsc** control command. MQSC commands can be entered interactively, by typing a command at the keyboard and waiting for the result. Alternatively, you can use a text editor to create a file that contains a sequence of MQSC commands and then run the file.

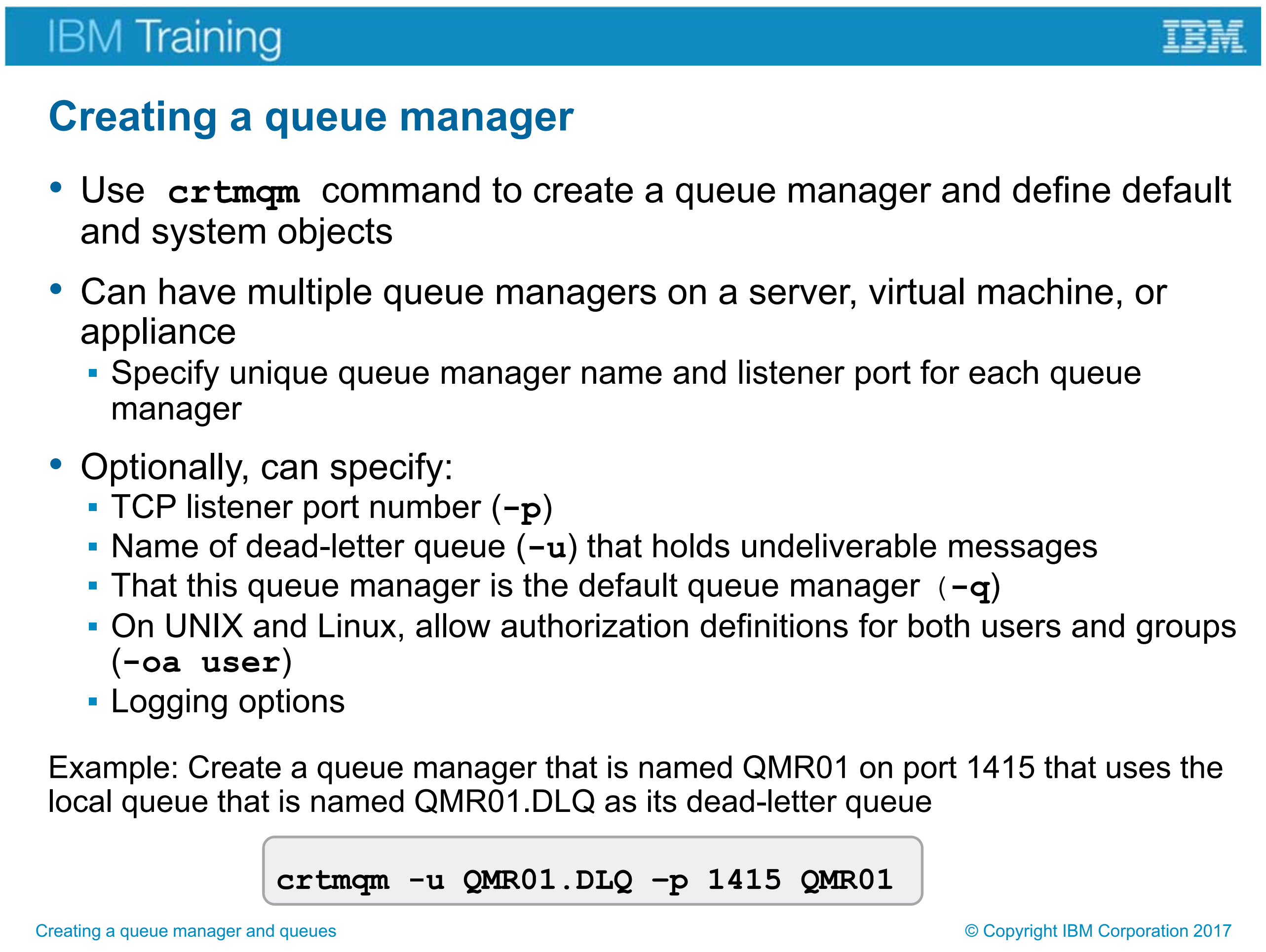
The figure shows examples of control commands and MQSC commands. In the example, the control command **runmqsc QMGR1** starts MQSC mode for the queue manager that is named

QMGR1. The commands in the MQSC mode are sent to QMGR1 until the **END** command is sent and the mode returns to control command mode.

Important

The operating system command prompt is not available when the system is in MQSC mode.

MQSC commands can be entered in uppercase or lowercase. In this course, MQSC commands are always shown in uppercase to help distinguish them from MQ control commands that are shown in lowercase.



IBM Training

Creating a queue manager

- Use **`crtmqm`** command to create a queue manager and define default and system objects
- Can have multiple queue managers on a server, virtual machine, or appliance
  - Specify unique queue manager name and listener port for each queue manager
- Optionally, can specify:
  - TCP listener port number (**`-p`**)
  - Name of dead-letter queue (**`-u`**) that holds undeliverable messages
  - That this queue manager is the default queue manager (**`-q`**)
  - On UNIX and Linux, allow authorization definitions for both users and groups (**`-oa user`**)
  - Logging options

Example: Create a queue manager that is named QMR01 on port 1415 that uses the local queue that is named QMR01.DLQ as its dead-letter queue

```
crtmqm -u QMR01.DLQ -p 1415 QMR01
```

Creating a queue manager and queues © Copyright IBM Corporation 2017

Figure 3-4. Creating a queue manager

After installation, the first MQ object to create is a queue manager by using the **`crtmqm`** control command. For the complete description of the **`crtmqm`** control command, see the IBM MQ product documentation.

Typically, you must create only one queue manager per system, but you can create multiple queue managers that are based on needs, such as security or separation of business units, testing, or development purposes.

Every queue manager has a name, which should be unique within a network of queue managers that exchange messages with each other. When a queue manager generates a unique message identifier for a message, it uses the first 12 characters of its name as part of the identifier.

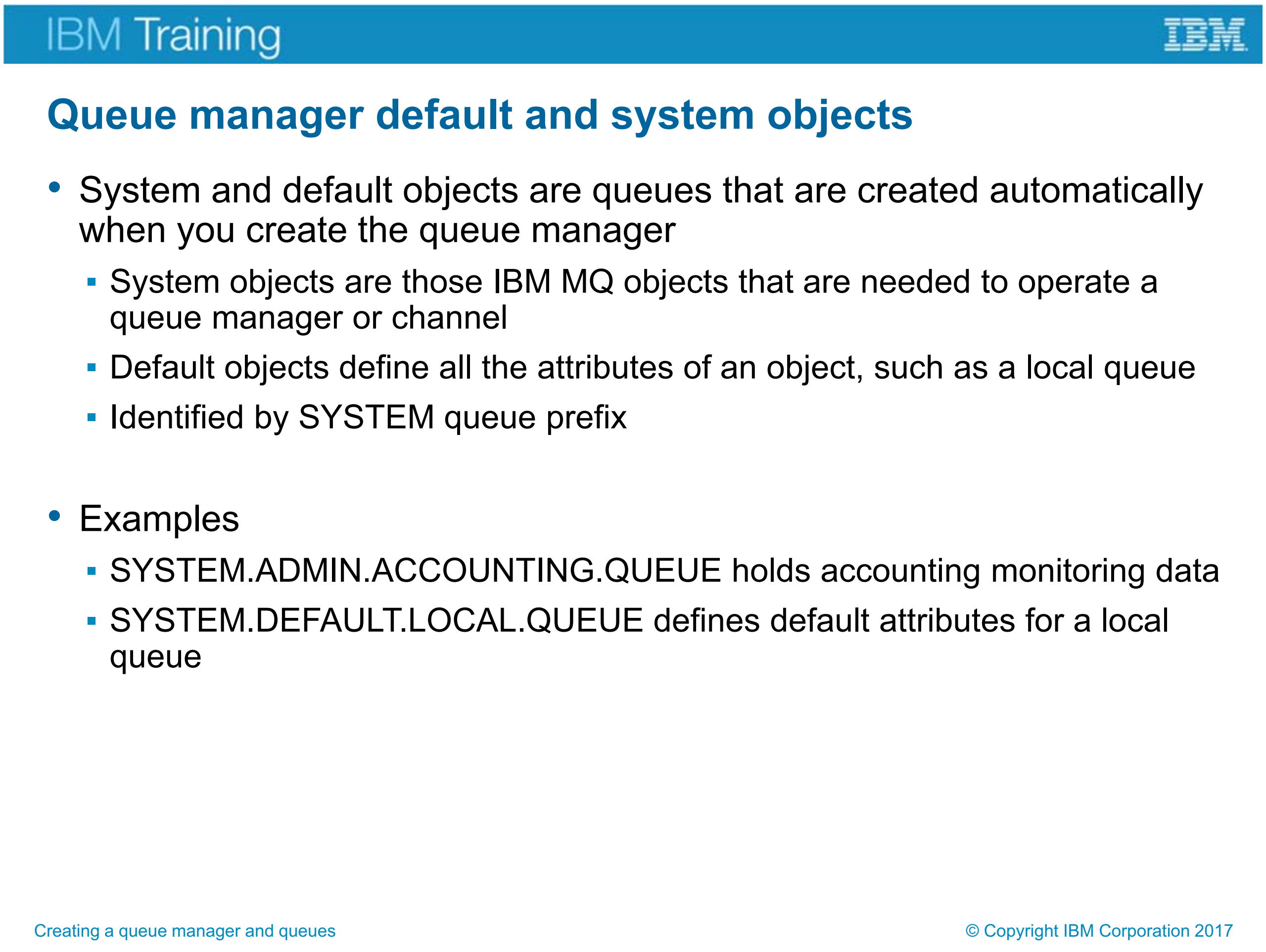
Queue manager names are case-sensitive. Always use uppercase characters to avoid problems that can be caused by using the incorrect case.

Plan conventions for queue manager names. For example, do not use host names as queue manager names when possible. If the host names change or frequent server reassignments occur, then the use of host names as queue names can lead to problems over time. Some large organizations use host names that are combined with Domain Name System (DNS) servers, which can decrease the configuration work that is required when queue managers move to different dedicated IBM MQ servers. Most organizations successfully use geographic area, the application that the MQ queue manager is serving, or a combination of the two.

You can define a dead-letter queue and a default transmission queue when you create the queue manager. You can also identify one queue manager on the system as the default queue manager.

If you are creating a queue manager for some simple tests, accept the default values for all the parameters unless a reason exists to use different ones.

You can modify some queue manager attributes later by using the **`ALTER QMGR`** MQSC command.



IBM Training

Queue manager default and system objects

- System and default objects are queues that are created automatically when you create the queue manager
  - System objects are those IBM MQ objects that are needed to operate a queue manager or channel
  - Default objects define all the attributes of an object, such as a local queue
  - Identified by SYSTEM queue prefix
- Examples
  - SYSTEM.ADMIN.ACCOUNTING.QUEUE holds accounting monitoring data
  - SYSTEM.DEFAULT.LOCAL.QUEUE defines default attributes for a local queue

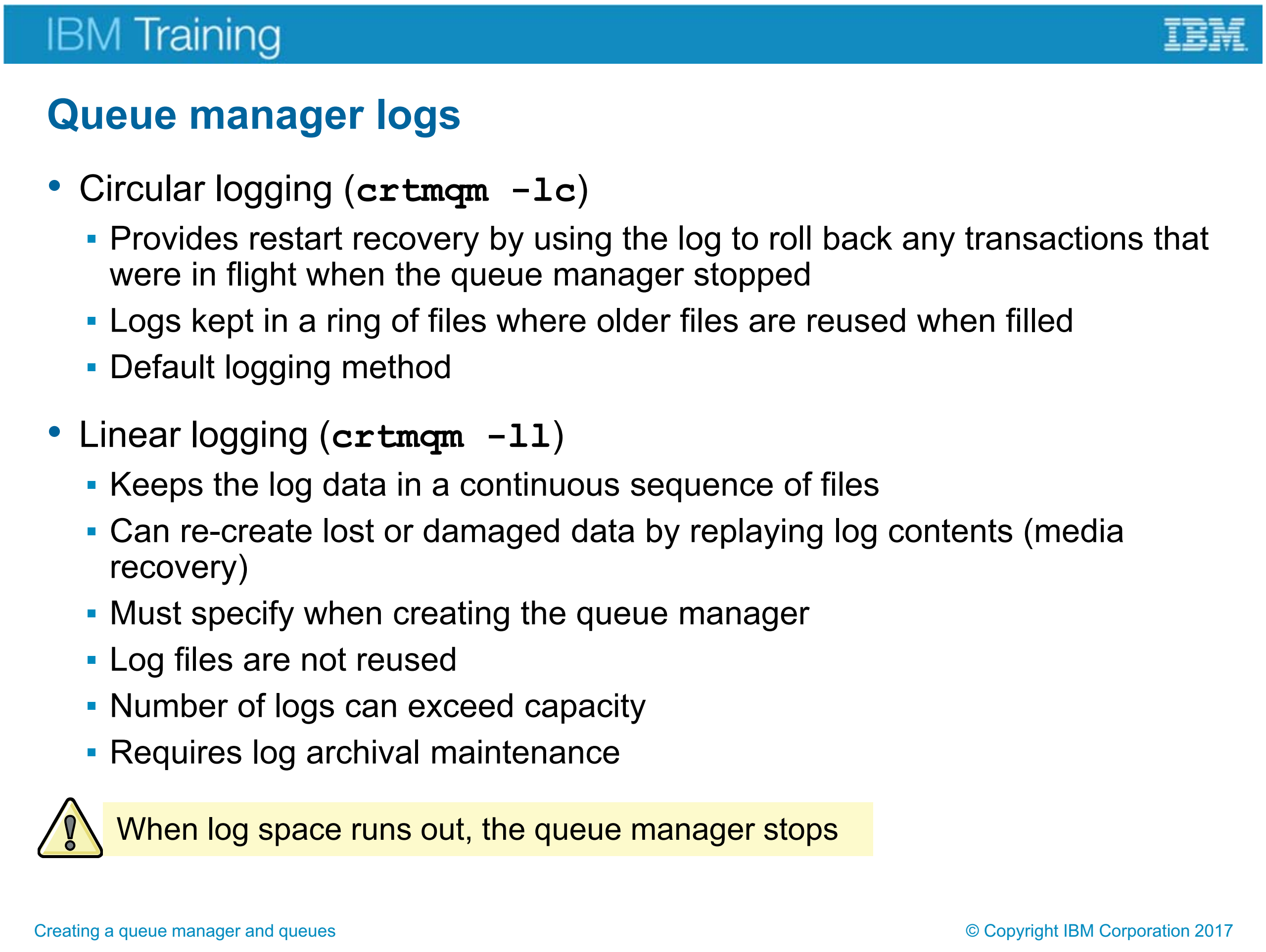
Creating a queue manager and queues

© Copyright IBM Corporation 2017

Figure 3-5. Queue manager default and system objects

The process of creating a queue manager also creates the processes, special queues that the queue manager needs for operation.

The queue manager queues are identified with the SYSTEM prefix. For example, the queue manager uses SYSTEM.ADMIN.ACCOUNTING.QUEUE to hold accounting monitoring data.



IBM Training

Queue manager logs

- Circular logging (`crtmqm -lc`)
  - Provides restart recovery by using the log to roll back any transactions that were in flight when the queue manager stopped
  - Logs kept in a ring of files where older files are reused when filled
  - Default logging method
- Linear logging (`crtmqm -ll`)
  - Keeps the log data in a continuous sequence of files
  - Can re-create lost or damaged data by replaying log contents (media recovery)
  - Must specify when creating the queue manager
  - Log files are not reused
  - Number of logs can exceed capacity
  - Requires log archival maintenance

When log space runs out, the queue manager stops

Creating a queue manager and queues © Copyright IBM Corporation 2017

Figure 3-6. Queue manager logs

MQ supports two ways of maintaining records of queue manager activities: circular logging and linear logging.

The type of logging and disk space to allocate are among the most important considerations to make when you create an MQ infrastructure.

Important

You cannot change the queue manager log type after the queue manager is created.

Circular logging keeps all restart data in a ring of log files. Logging fills the first file in the ring, then moves on to the next, until all the files are full. It then goes back to the first file in the ring and starts again. This process continues while the product is in use, and has the advantage that you never run out of log files. When a queue manager is created on a distributed operating system, the default log type is circular.

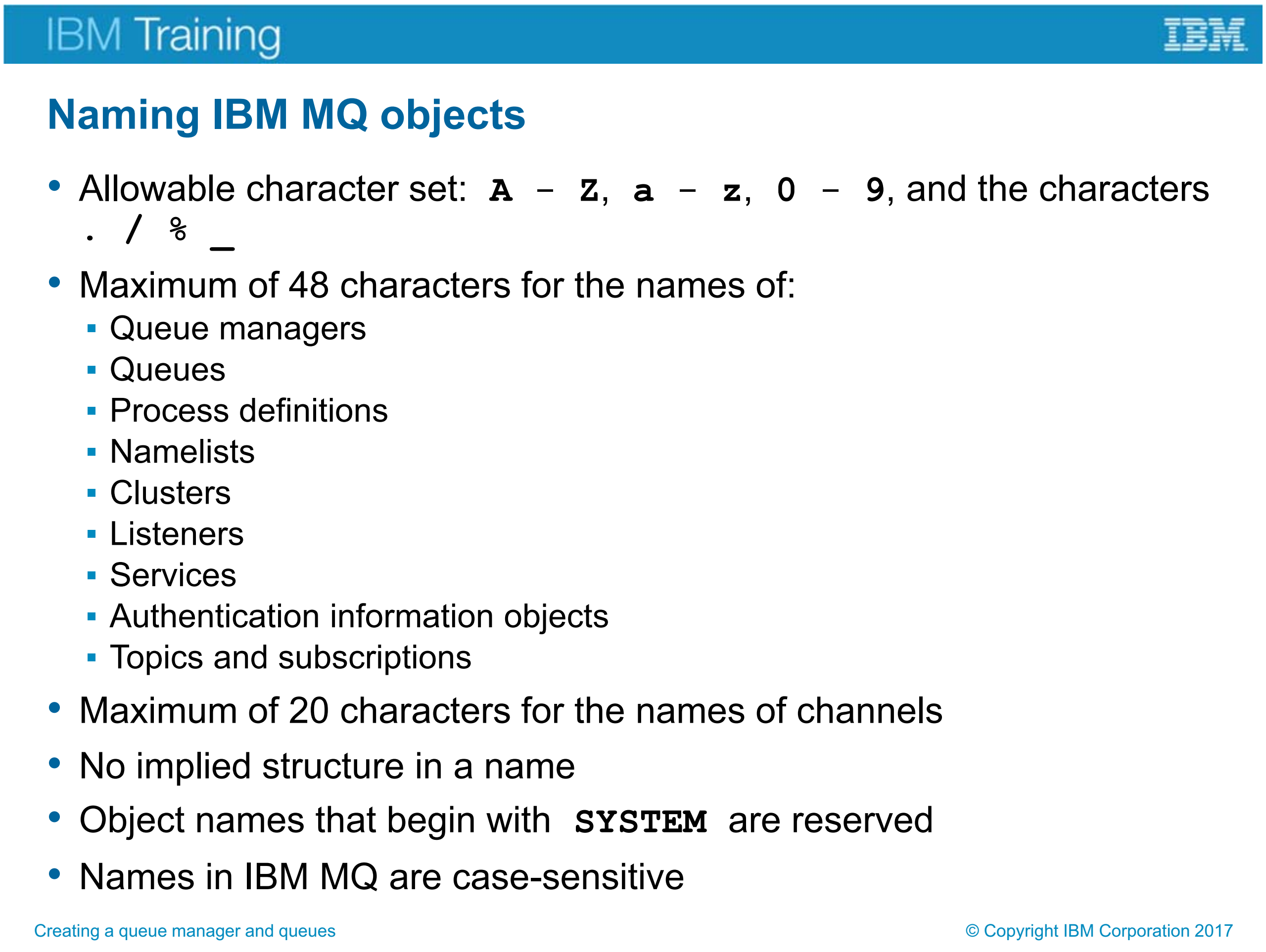
Linear logging keeps the log data in a continuous sequence of files. Space is not reused, so you can always retrieve any record that is logged in any log extent that was not deleted. The number of log files that are used with linear logging can be large, depending on your message flow and the age of your queue manager. As disk space is finite, you might have to think about some form of archiving.

If you want both restart recovery and media recovery to re-create lost or damaged data by replaying the contents of the log, use linear logging. When linear logging is used, MQ tracks which logs are no longer necessary. However, MQ does not discard these files. The log maintenance must be implemented as another automated system process that uses the operating system-specific file utilities.

Important

Linear logging requires regular maintenance to discard logs that are no longer needed. If log maintenance is not done, log space is exhausted and the queue manager stops.

An organization should always establish standards on how to configure logging for a queue manager. Organizations also need to institute automated methods and scripts to maintain the logs.



IBM Training

Naming IBM MQ objects

- Allowable character set: `A - Z`, `a - z`, `0 - 9`, and the characters `. / % _`
- Maximum of 48 characters for the names of:
  - Queue managers
  - Queues
  - Process definitions
  - Namelists
  - Clusters
  - Listeners
  - Services
  - Authentication information objects
  - Topics and subscriptions
- Maximum of 20 characters for the names of channels
- No implied structure in a name
- Object names that begin with `SYSTEM` are reserved
- Names in IBM MQ are case-sensitive

Creating a queue manager and queues © Copyright IBM Corporation 2017

Figure 3-7. Naming IBM MQ objects

The figure lists some of the rules for naming MQ objects such as queue managers and queues.

In general, MQ object names are limited to 48 characters. Period, forward slash, percent sign, and underscore are special characters that are allowed in MQ object names. National language characters are not allowed.

Names can be enclosed in double quotation marks. If you use the forward slash or percent sign characters in a name, the name must be enclosed in double quotation marks.

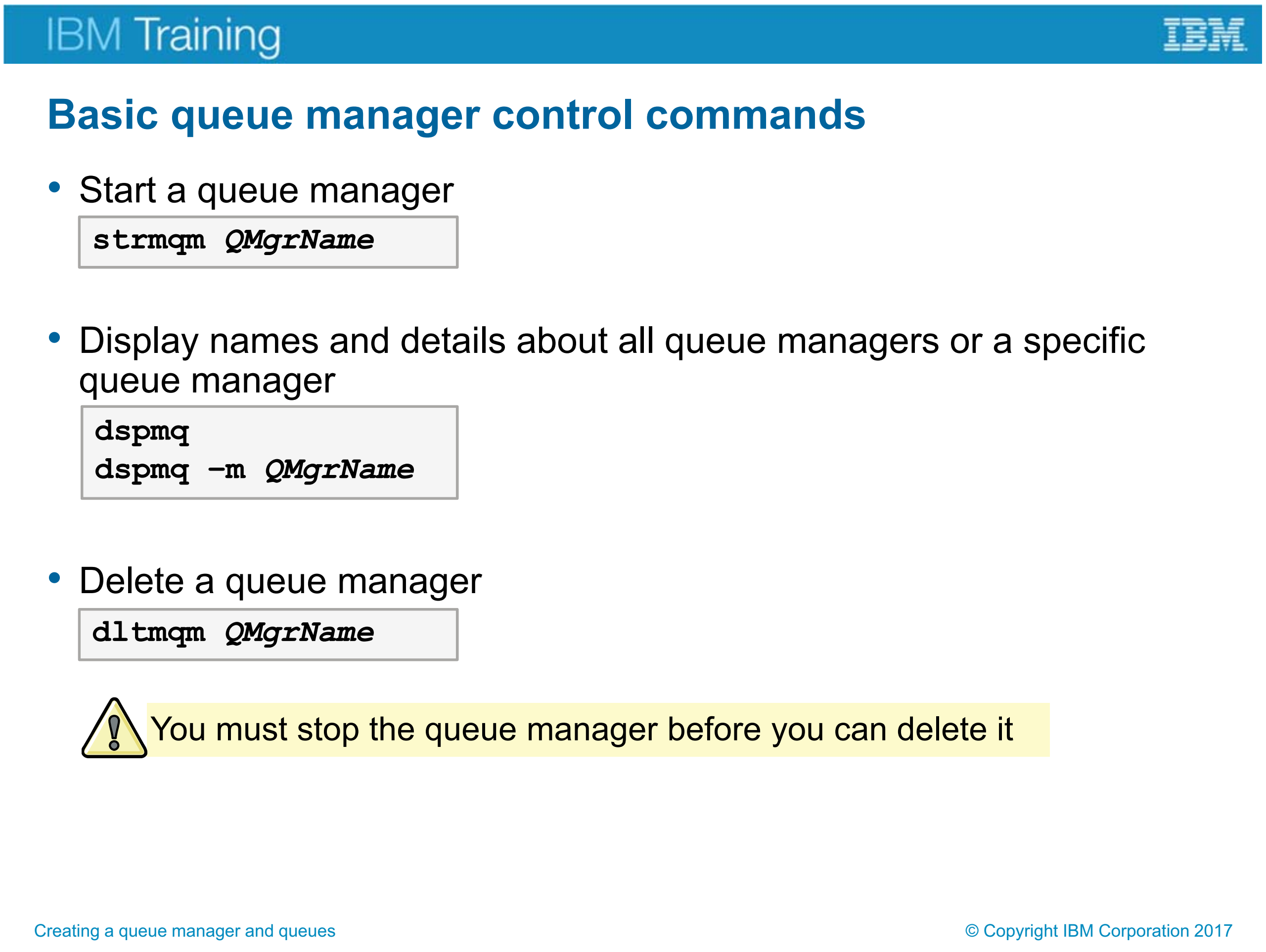
Leading or embedded blanks are not allowed in MQ object names.

Channel names are limited to 20 characters but otherwise follow the standard rules for naming MQ objects.

The rules for naming MQ objects are the same on all supported operating systems. No implied structure exists in a name that you might find in the rules for file names on many operating systems.

Agree on all names before you start.

Names are case-sensitive. Be consistent in using uppercase and lowercase characters. Many organizations use uppercase characters for names, especially when z/OS queue managers are used within the MQ environment.



IBM Training

Basic queue manager control commands

- Start a queue manager

```
strmqm QMgrName
```

- Display names and details about all queue managers or a specific queue manager

```
dspmq
dspmq -m QMgrName
```

- Delete a queue manager

```
dltmqm QMgrName
```

You must stop the queue manager before you can delete it

Creating a queue manager and queues

Figure 3-8. Basic queue manager control commands

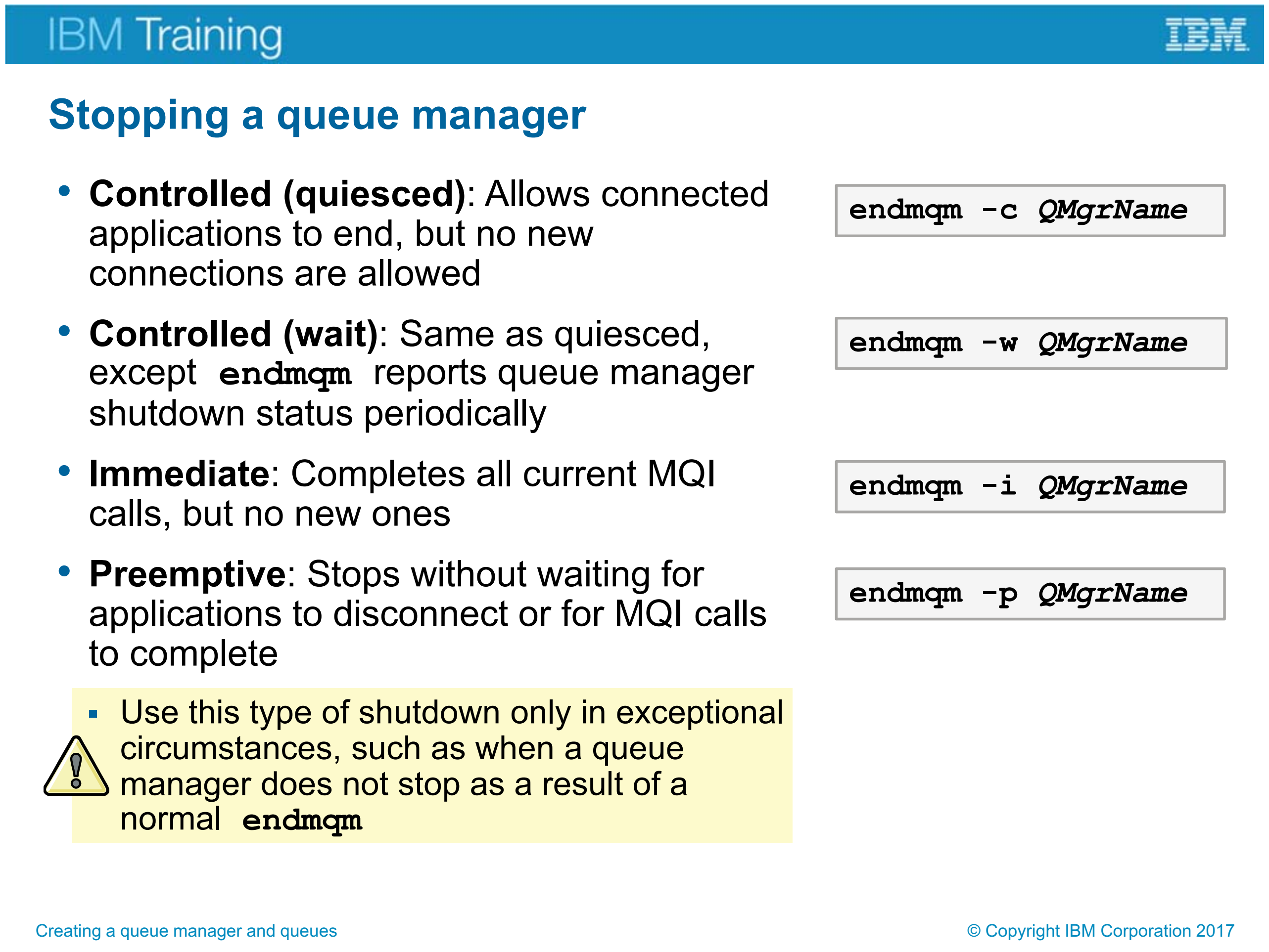
When a queue manager is first created, it is created in a *stopped* state. To start the queue manager, use the **`strmqm`** command followed by the name of the queue manager.

To display the status and queue manager configuration information, use the **`dspmq`** command.

To delete a queue manager and its associated objects, use the **`dltmqm`** command. You must stop the queue manager before you can delete it.

The create queue manager command, **`crtmqm`**, and the commands that are listed in this figure are control commands. The name of the queue manager is a required parameter on some of these commands. It is a good practice to always include the queue manager name in the command to ensure that the command is run against the correct queue manager.

The control commands are described in IBM MQ product documentation.



IBM Training

Stopping a queue manager

- **Controlled (quiesced)**: Allows connected applications to end, but no new connections are allowed

```
endmqm -c QMgrName
```

- **Controlled (wait)**: Same as quiesced, except **`endmqm`** reports queue manager shutdown status periodically

```
endmqm -w QMgrName
```

- **Immediate**: Completes all current MQI calls, but no new ones

```
endmqm -i QMgrName
```

- **Preemptive**: Stops without waiting for applications to disconnect or for MQI calls to complete
  - Use this type of shutdown only in exceptional circumstances, such as when a queue manager does not stop as a result of a normal **`endmqm`**

```
endmqm -p QMgrName
```

Creating a queue manager and queues © Copyright IBM Corporation 2017

Figure 3-9. Stopping a queue manager

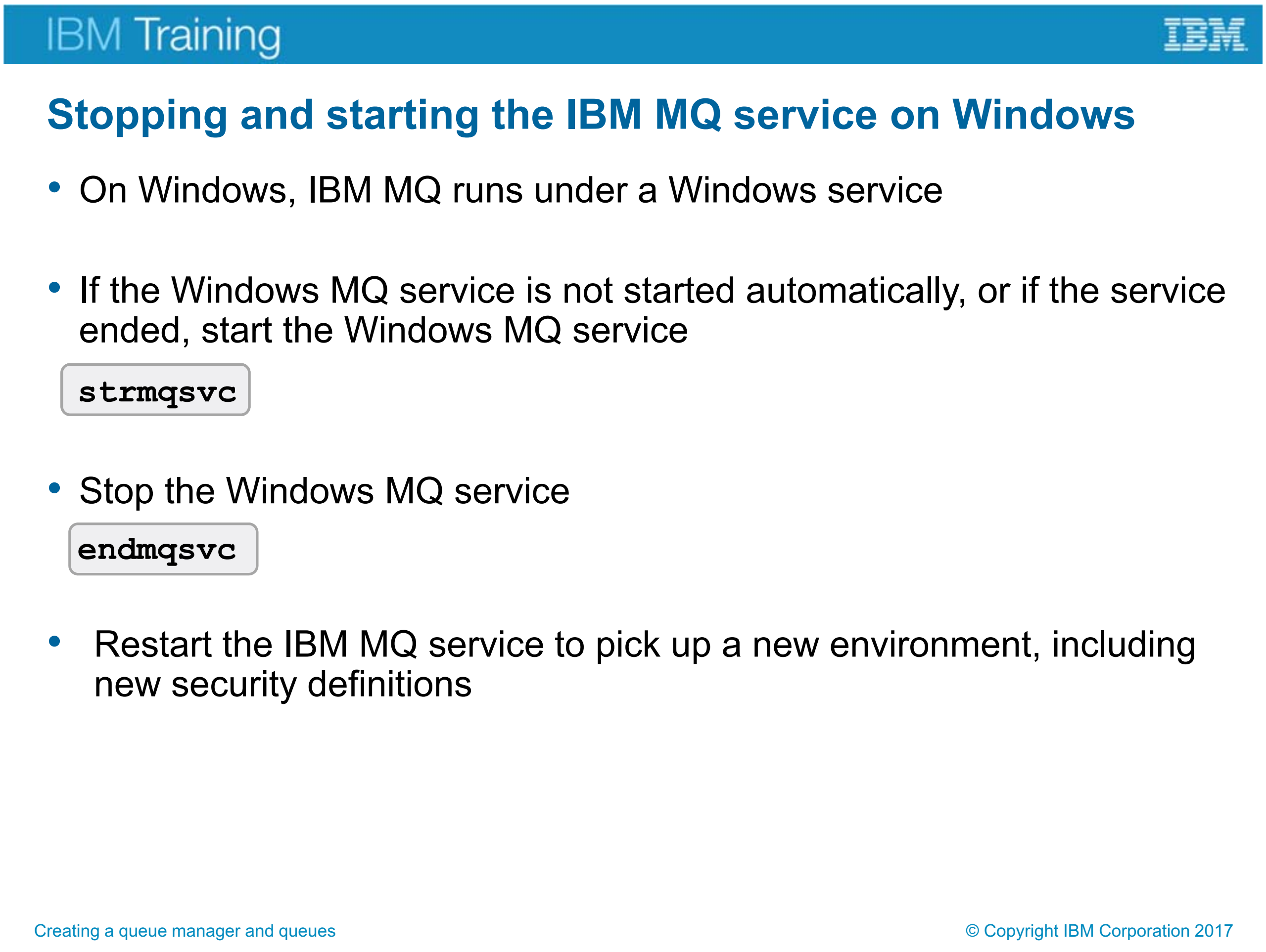
The control command to stop the queue manager is **endmqm**. Four options exist for controlling how the queue manager shuts down.

- **Controlled (or quiesced) shutdown**: The queue manager stops after all applications are disconnected. All new requests to connect to the queue manager fail. Controlled shutdown is the default mode.
- **Controlled shut down with wait**: End programs in the same manner as the controlled option. However, the command prompt does not return until the queue manager stops.
- **Immediate shut down**: The queue manager stops after it completes all the in-process MQI calls. Any MQI calls that are sent after this command is entered fail. Any incomplete units of work are rolled back when the queue manager is next started.
- **Preemptive shut down**: The queue manager stops without waiting for applications to disconnect or for MQI calls to complete. Use of this mode can lead to unpredictable results and should be used as the last option.

The normal mode of stopping a queue manager is the controlled, or quiesced, mode. In this mode, applications can continue to do work, but well-behaved applications disconnect as soon as it is convenient.

Use the immediate mode of stopping a queue manager only if you need to stop it quickly with predictable results.

Use the preemptive mode only if all other options to stop the queue manager fail.



IBM Training

Stopping and starting the IBM MQ service on Windows

- On Windows, IBM MQ runs under a Windows service
- If the Windows MQ service is not started automatically, or if the service ended, start the Windows MQ service

```
strmqsvc
```

- Stop the Windows MQ service

```
endmqsvc
```

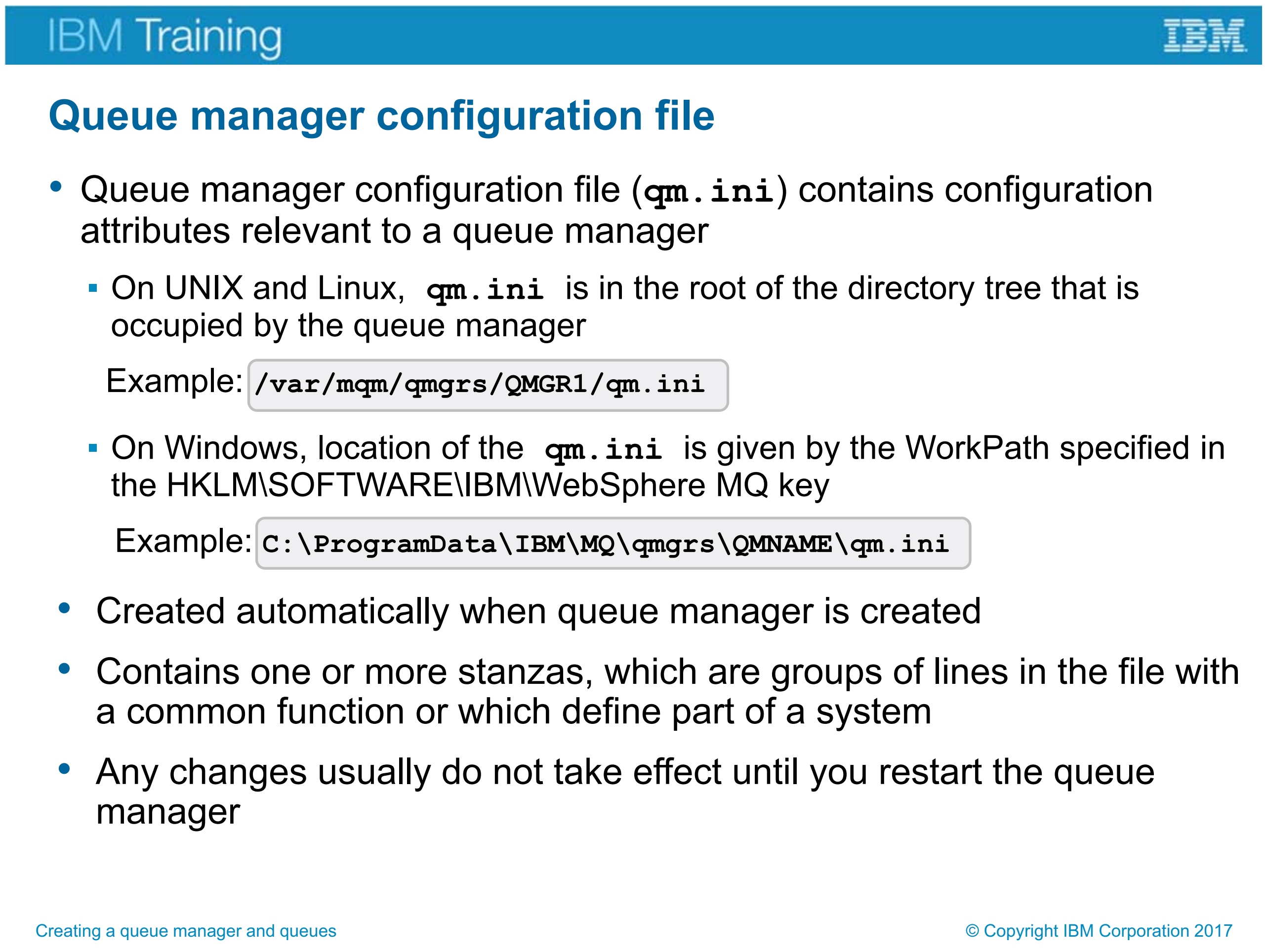
- Restart the IBM MQ service to pick up a new environment, including new security definitions

Creating a queue manager and queues © Copyright IBM Corporation 2017

Figure 3-10. Stopping and starting the IBM MQ service on Windows

MQ runs as a service on Windows. It might be necessary to restart the MQ service so that MQ recognizes a new environment.

You can restart the MQ service on Windows by using the Services Administrative Tool or by using the MQ control commands **`endmqsvc`** and **`strmqsvc`**.



IBM Training

Queue manager configuration file

- Queue manager configuration file (**`qm.ini`**) contains configuration attributes relevant to a queue manager
  - On UNIX and Linux, **`qm.ini`** is in the root of the directory tree that is occupied by the queue manager

    Example: `/var/mqm/qmgrs/QMGR1/qm.ini`
  - On Windows, location of the **`qm.ini`** is given by the WorkPath specified in the HKLM\SOFTWARE\IBM\WebSphere MQ key

    Example: `C:\ProgramData\IBM\MQ\qmgrs\QMNAME\qm.ini`
- Created automatically when queue manager is created
- Contains one or more stanzas, which are groups of lines in the file with a common function or which define part of a system
- Any changes usually do not take effect until you restart the queue manager

Creating a queue manager and queues © Copyright IBM Corporation 2017

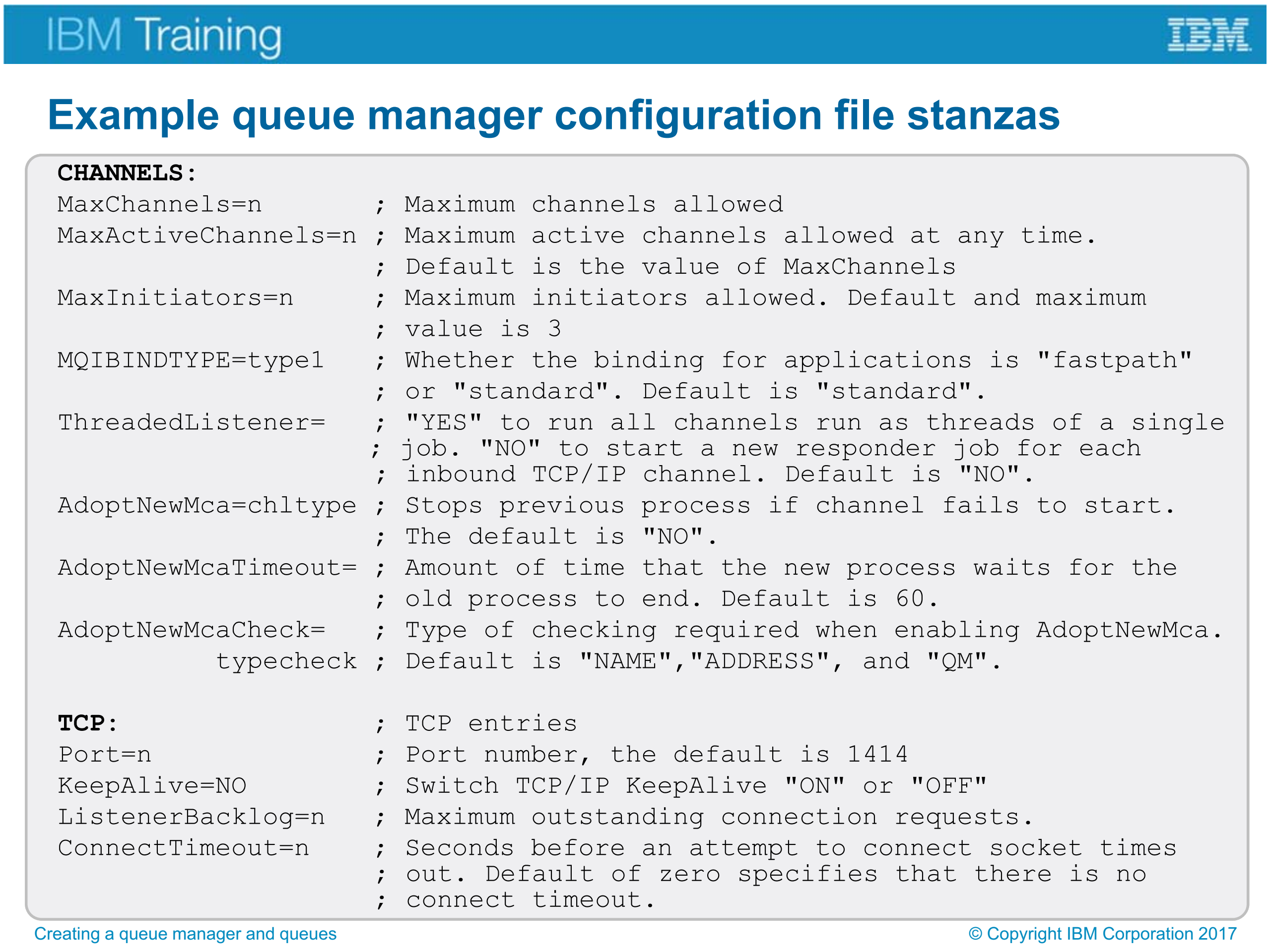
Figure 3-11. Queue manager configuration file

When a queue manager is created, a queue manager configuration file that is named **`qm.ini`** is automatically created at the same time. This file contains information that is relevant to a specific queue manager. One queue manager configuration file exists for each queue manager.

The queue manager configuration file contains texts and is readable. The contents of the file might change by certain commands and in special circumstances.

The queue manager configuration file contains configuration parameters that are grouped by function into stanzas.

Most changes to the queue manager configuration file are not recognized until the queue manager restarts.



IBM Training

Example queue manager configuration file stanzas

```
CHANNELS:
MaxChannels=n        ; Maximum channels allowed
MaxActiveChannels=n  ; Maximum active channels allowed at any time.
                     ; Default is the value of MaxChannels
MaxInitiators=n      ; Maximum initiators allowed. Default and maximum
                     ; value is 3
MQIBINDTYPE=type1    ; Whether the binding for applications is "fastpath"
                     ; or "standard". Default is "standard".
ThreadedListener=    ; "YES" to run all channels run as threads of a single
                     ; job. "NO" to start a new responder job for each
                     ; inbound TCP/IP channel. Default is "NO".
AdoptNewMca=chltype  ; Stops previous process if channel fails to start.
                     ; The default is "NO".
AdoptNewMcaTimeout=  ; Amount of time that the new process waits for the
                     ; old process to end. Default is 60.
AdoptNewMcaCheck=    ; Type of checking required when enabling AdoptNewMca.
          typecheck  ; Default is "NAME","ADDRESS", and "QM".

TCP:                 ; TCP entries
Port=n               ; Port number, the default is 1414
KeepAlive=NO         ; Switch TCP/IP KeepAlive "ON" or "OFF"
ListenerBacklog=n    ; Maximum outstanding connection requests.
ConnectTimeout=n     ; Seconds before an attempt to connect socket times
                     ; out. Default of zero specifies that there is no
                     ; connect timeout.
```

Creating a queue manager and queues © Copyright IBM Corporation 2017

Figure 3-12. Example queue manager configuration file stanzas

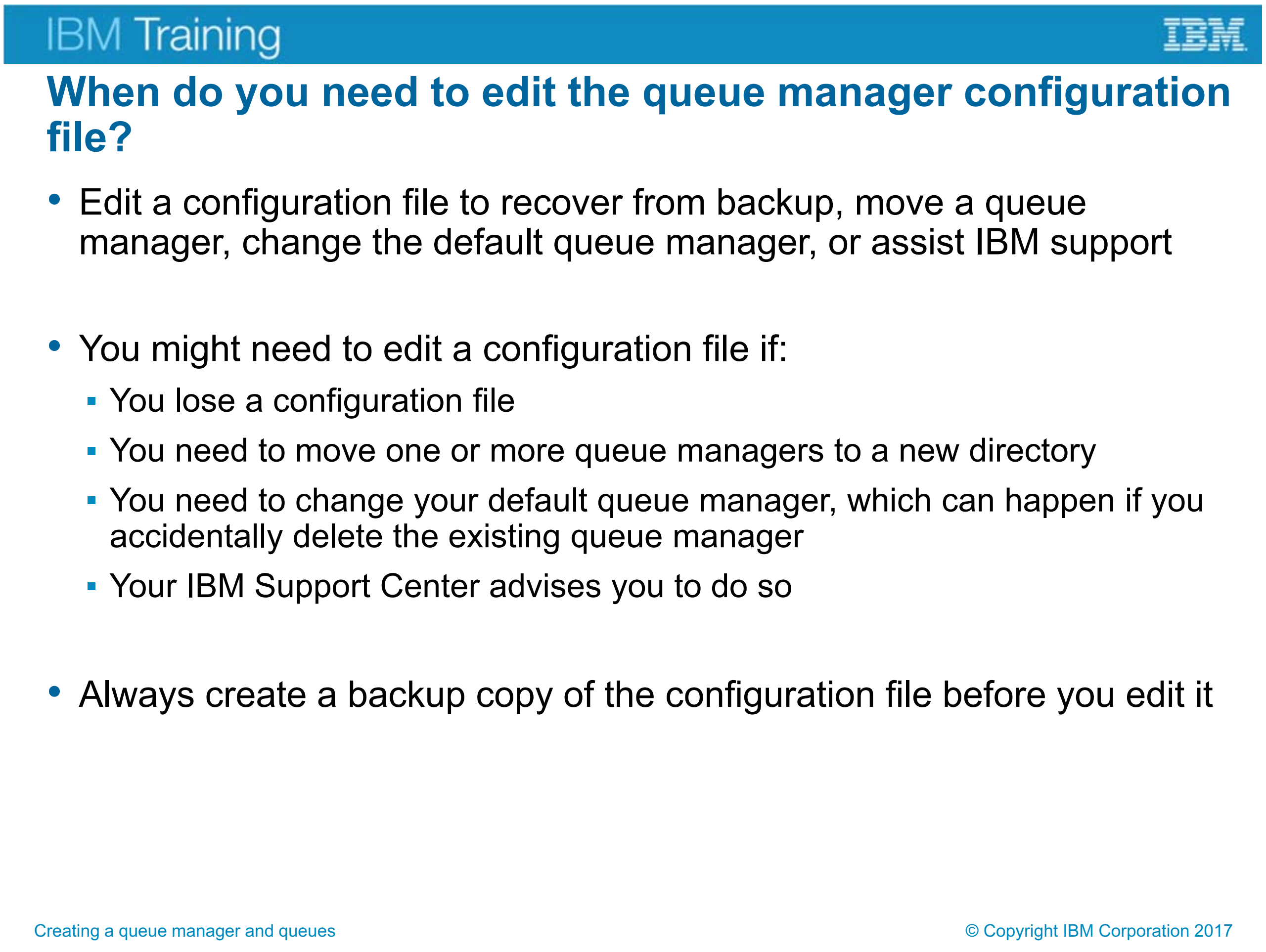
This figure shows the CHANNELS and TCP stanzas in the queue manager configuration file.

You can modify the **`qm.ini`** file, use MQSC commands, or use the queue manager **Properties** pages in MQ Explorer to specify channel and network protocol configuration parameters.

Note

Only attributes that represent changes to the default values need to be specified in the queue manager configuration file.

For more information about the queue manager configuration file, see the IBM MQ product documentation.



IBM Training

When do you need to edit the queue manager configuration file?

- Edit a configuration file to recover from backup, move a queue manager, change the default queue manager, or assist IBM support
- You might need to edit a configuration file if:
  - You lose a configuration file
  - You need to move one or more queue managers to a new directory
  - You need to change your default queue manager, which can happen if you accidentally delete the existing queue manager
  - Your IBM Support Center advises you to do so
- Always create a backup copy of the configuration file before you edit it

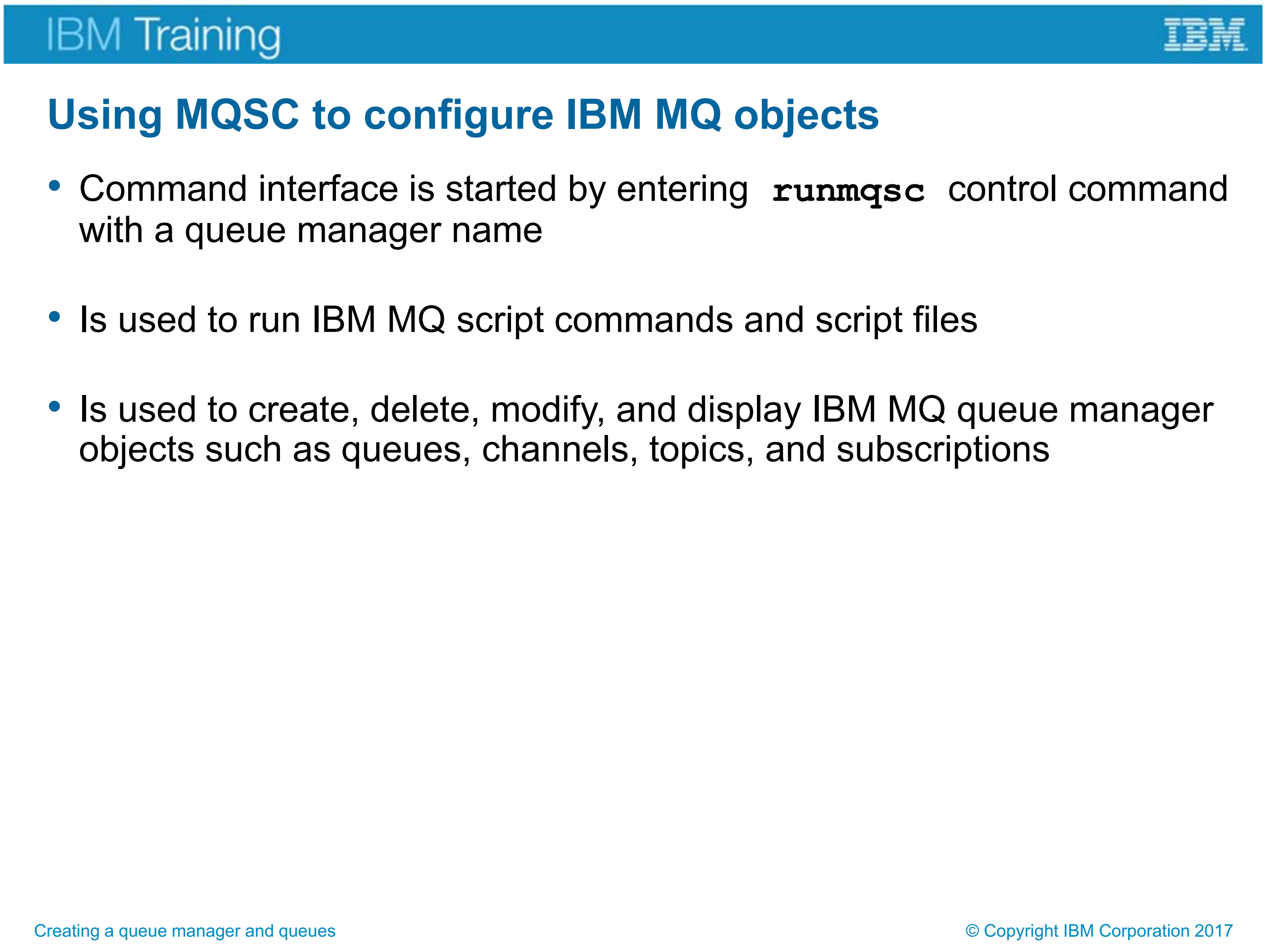
Creating a queue manager and queues © Copyright IBM Corporation 2017

Figure 3-13. When do you need to edit the queue manager configuration file?

Specific events might require that you edit the queue manager configuration file.

For example, it might be necessary to edit a queue manager configuration file to recover from a backup or to assist IBM support. This figure lists some other reasons why it might be necessary to edit the queue manager configuration file.

Be sure to always create a backup copy of the queue manager configuration file before you edit it.



IBM Training

Using MQSC to configure IBM MQ objects

- Command interface is started by entering **`runmqsc`** control command with a queue manager name
- Is used to run IBM MQ script commands and script files
- Is used to create, delete, modify, and display IBM MQ queue manager objects such as queues, channels, topics, and subscriptions

Creating a queue manager and queues © Copyright IBM Corporation 2017

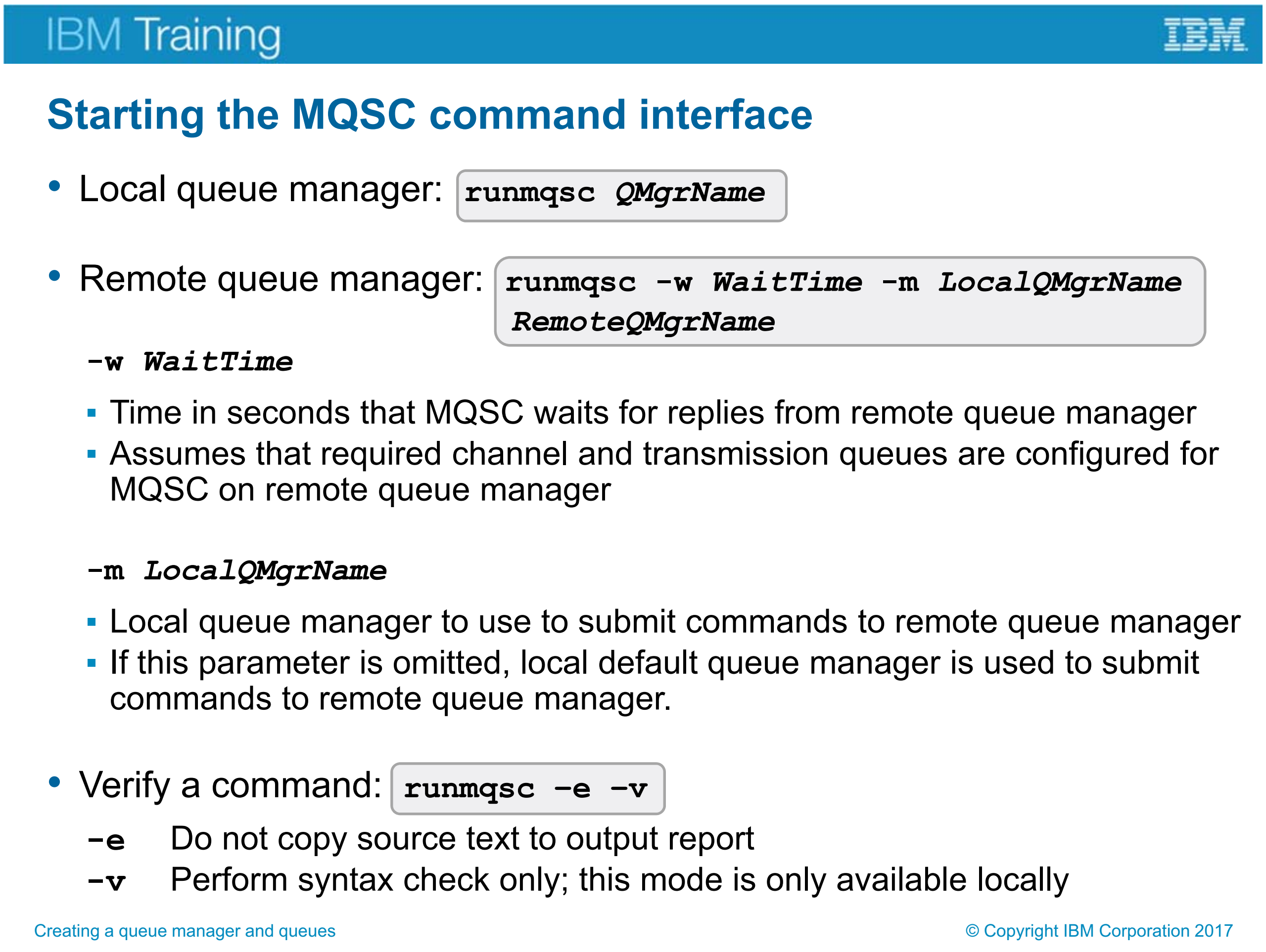
Figure 3-14. Using MQSC to configure IBM MQ objects

On all queue managers, the administration interface for creating, modifying, deleting, and displaying queue manager resources and objects is MQSC.

MQSC commands are known as script commands because they are frequently batched together to create a repeatable set of MQ resource definitions in a script file.

MQSC commands are also used to display the runtime status of objects like channels.

When you enter MQSC commands interactively, you can get help by typing the name of the command and a question mark character: command?



IBM Training

Starting the MQSC command interface

- Local queue manager: `runmqsc QMgrName`
- Remote queue manager: `runmqsc -w WaitTime -m LocalQMgrName RemoteQMgrName`

  `-w WaitTime`
  - Time in seconds that MQSC waits for replies from remote queue manager
  - Assumes that required channel and transmission queues are configured for MQSC on remote queue manager

  `-m LocalQMgrName`
  - Local queue manager to use to submit commands to remote queue manager
  - If this parameter is omitted, local default queue manager is used to submit commands to remote queue manager.
- Verify a command: `runmqsc -e -v`

  `-e` Do not copy source text to output report

  `-v` Perform syntax check only; this mode is only available locally

Creating a queue manager and queues © Copyright IBM Corporation 2017

Figure 3-15. Starting the MQSC command interface

The command for starting MQSC mode is **runmqsc** followed by the name of a local queue manager.

If you are trying to connect to a remote queue manager, you must specify a local queue manager that can be used to submit commands to the remote queue manager. If the remote queue manager is on z/OS, then the `-x` option is required.

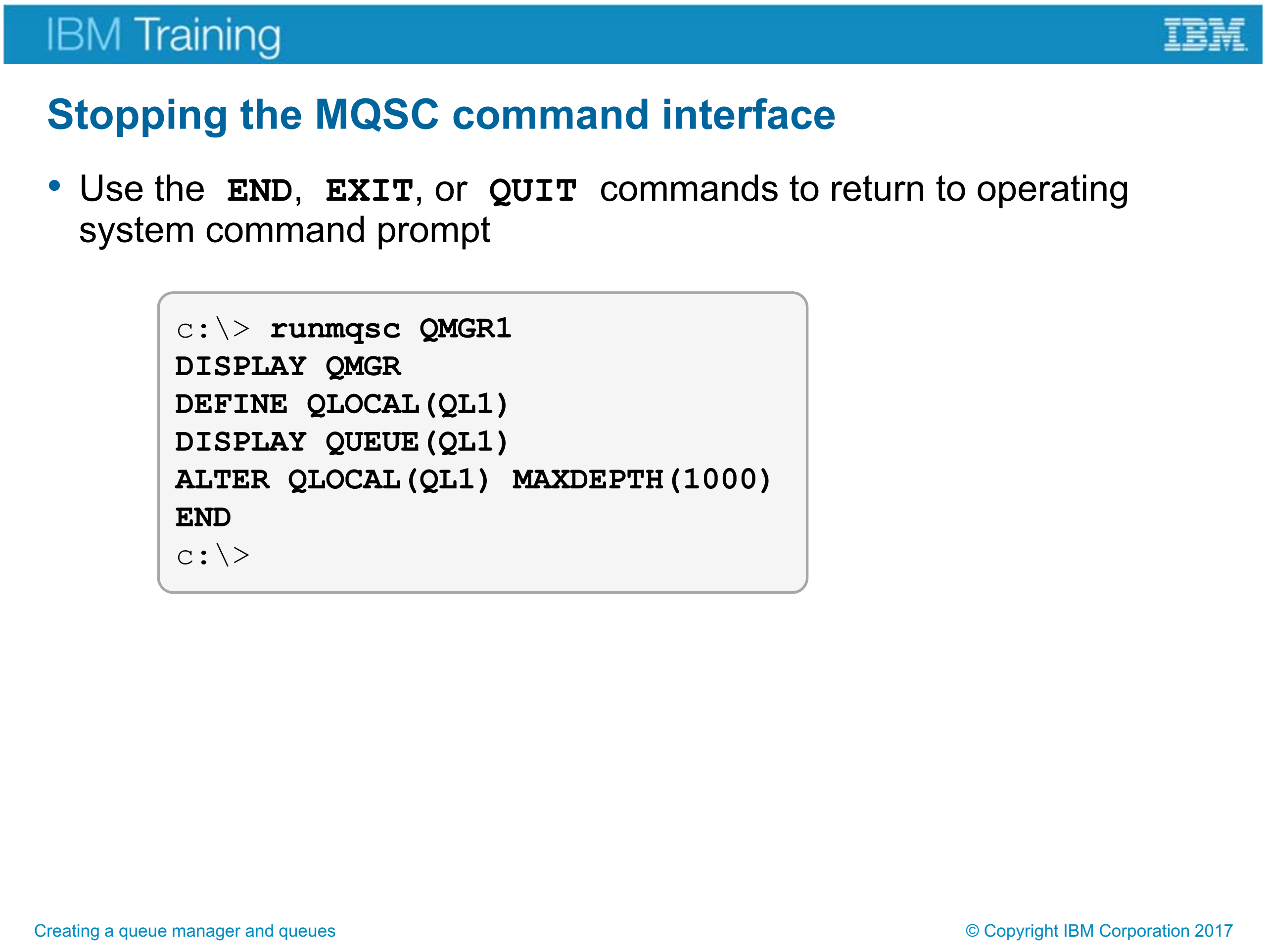
The input to **runmqsc** is zero, one, or more MQ script commands, and the output is the result of running those commands, including operator and error messages. Alternatively, you can create a file that contains a sequence of MQSC commands and then have them run with the results that are directed to a file.

MQSC has three modes of operation:

- **Verify (-v)**: Input commands are read and checked but not run.
- **Direct**: Connect to a local queue manager and do not use any intermediate queues or channels.
- **Indirect**: Connect to a remote queue manager. For indirect connections, the appropriate channels and transmission queues must be established. This mode allows MQSC to connect and administer a remote queue manager if security is configured to allow administration.

The **`-e`** option on the **`runmqsc`** command prevents source text for the MQSC commands from being copied into a report. This option is useful when you enter commands interactively.

The **`-w`** option on the **`runmqsc`** command is used for indirect connections and specifies the time, in seconds, that MQSC waits for replies. Any replies that are received after this time are discarded, but the MQSC commands still run.



IBM Training

Stopping the MQSC command interface

- Use the **END**, **EXIT**, or **QUIT** commands to return to operating system command prompt

```
c:\> runmqsc QMGR1
DISPLAY QMGR
DEFINE QLOCAL(QL1)
DISPLAY QUEUE(QL1)
ALTER QLOCAL(QL1) MAXDEPTH(1000)
END
c:\>
```

Creating a queue manager and queues © Copyright IBM Corporation 2017

Figure 3-16. Stopping the MQSC command interface

To exit MQSC mode and return to control command mode and the operating system prompt, enter **END**, **EXIT**, or **QUIT**. Optionally, you can enter the end-of-file (EOF) character for the operating system:

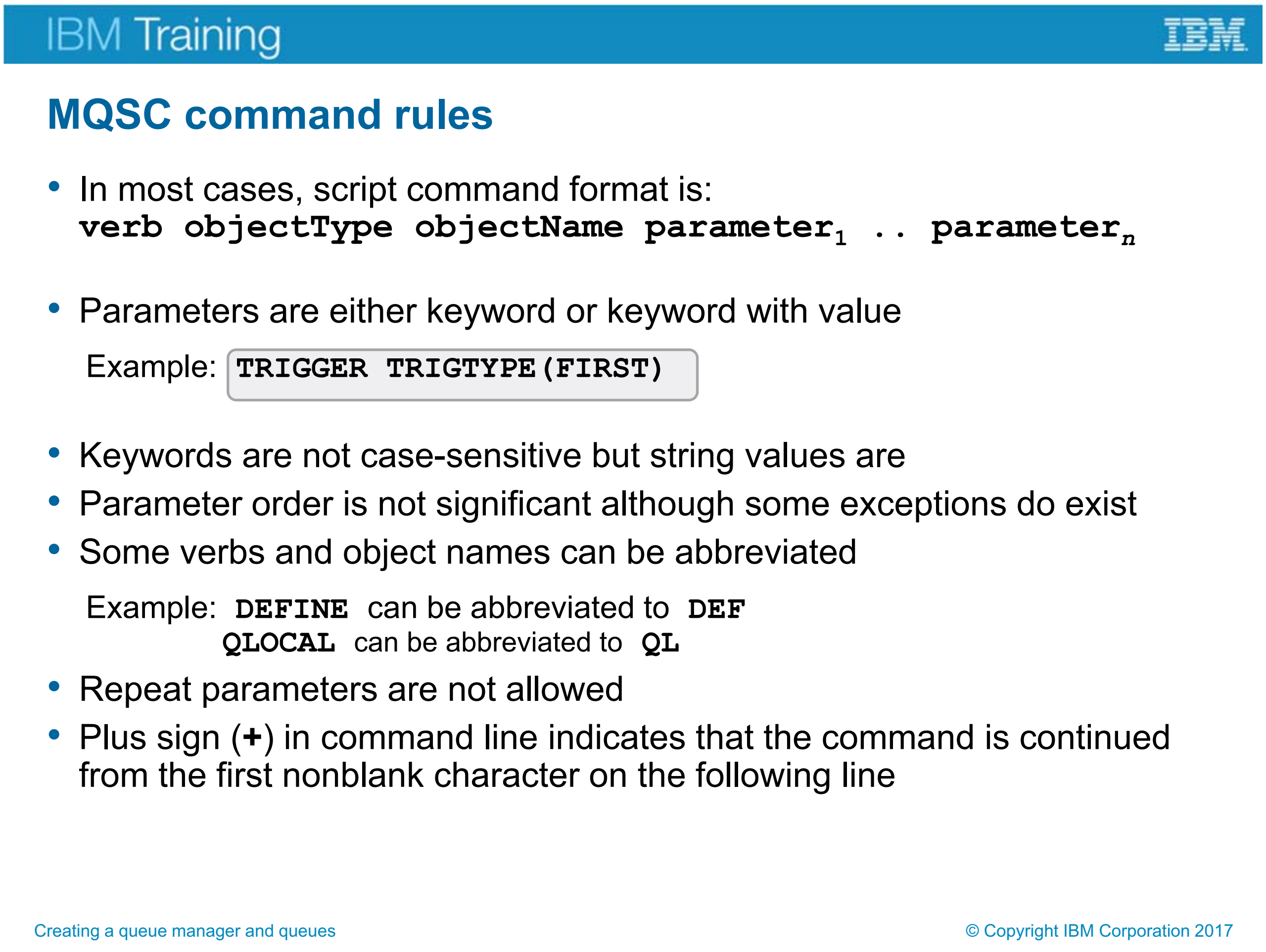
- On Windows, type Ctrl+Z, and press **Enter**.
- On UNIX or Linux, type Ctrl+D.

The example in the figure shows the control command that starts MQSC mode for the queue manager that is named QMGR1: **`runmqsc QMGR1`**

The next four statements are MQSC commands:

- **`DISPLAY QMGR`** displays information about the queue manager QMGR1.
- **`DEFINE QLOCAL(QL1)`** defines a local queue that is named **QL1**.
- **`DISPLAY QUEUE(QL1)`** displays information about the local queue that is named **QL1**.
- **`ALTER QLOCAL(QL1) MAXDEPTH(1000)`** changes the **MAXDEPTH** property for the local queue that is named **QL1**.

The **`END`** statement stops MQSC mode and returns control to the operating system.


IBM Training

MQSC command rules

- In most cases, script command format is:
  `verb objectType objectName parameter`$_1$ **`..`** **`parameter`**$_n$
- Parameters are either keyword or keyword with value

  Example: **`TRIGGER TRIGTYPE(FIRST)`**
- Keywords are not case-sensitive but string values are
- Parameter order is not significant although some exceptions do exist
- Some verbs and object names can be abbreviated

  Example: **`DEFINE`** can be abbreviated to **`DEF`**
  `QLOCAL` can be abbreviated to **`QL`**
- Repeat parameters are not allowed
- Plus sign (**+**) in command line indicates that the command is continued from the first nonblank character on the following line

Creating a queue manager and queues © Copyright IBM Corporation 2017

Figure 3-17. MQSC command rules

The figure lists some of the rules for the specification of MQSC commands.

The command verb must always come first and is followed by, in most cases, an MQ object, and then optional keyword and keyword/value parameters.

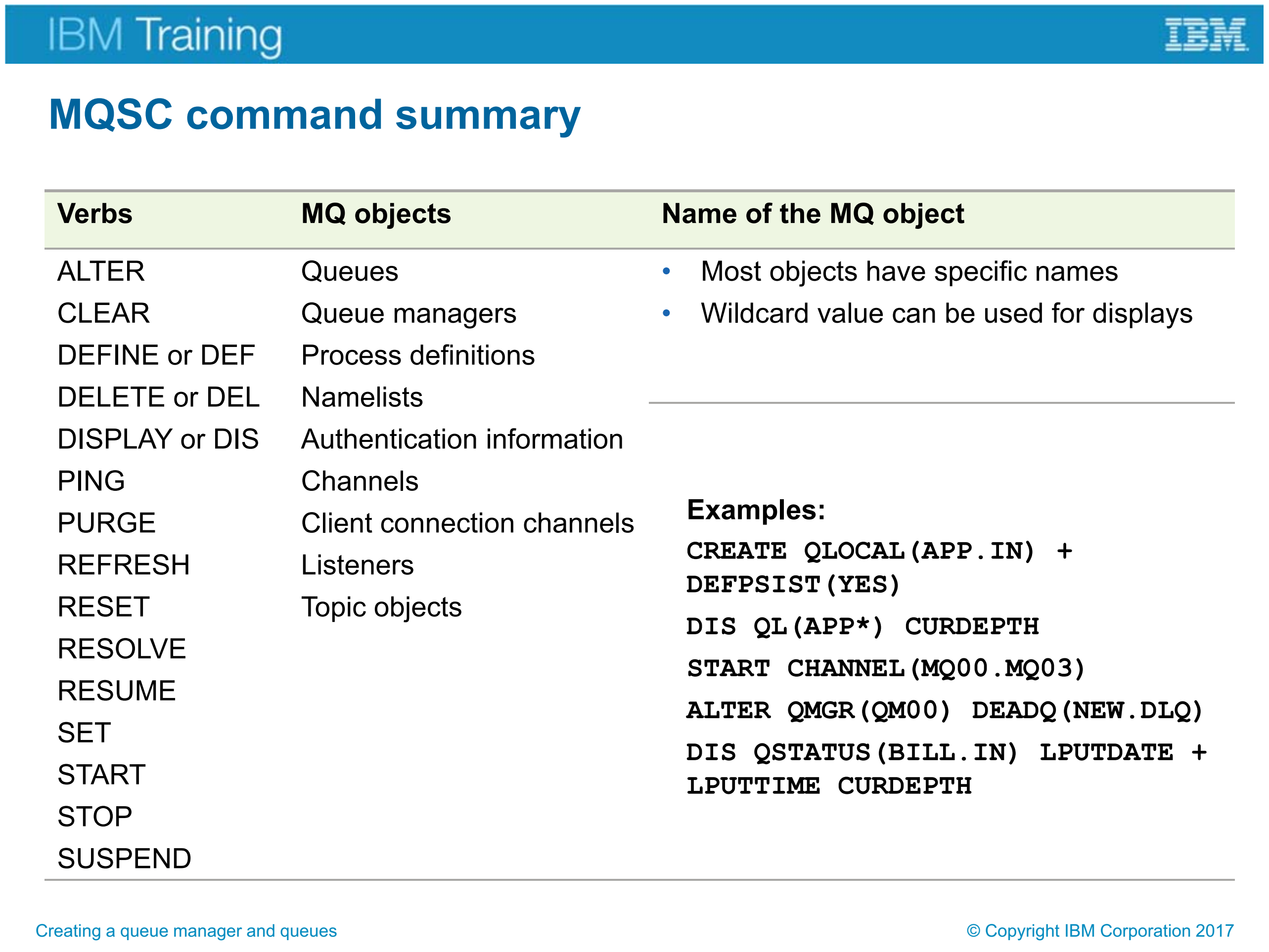
Script commands are not case-sensitive, but names and parameter values are case-sensitive.

In most cases, parameter order is not important, and some verbs can be abbreviated. An exception to parameter order is that the parameter **`CHLTYPE`** must be the first parameter that is specified in **`DEFINE CHANNEL`** and **`ALTER CHANNEL`** commands on distributed operating systems.

Abbreviations are fixed and are described in the MQ product documentation under their relevant MQSC command entries as synonyms. For example, the synonym for **`REFRESH CLUSTER`** is **`REF CLUSTER`**. The synonym for **`DEFINE QLOCAL()`** is **`DEF QL()`**.

Parameters cannot be repeated within the same script command. For example, **`ALTER QLOCAL(ANDREW) TRIGGER TRIGGER`** is not valid. Repeating the negation of the same parameter, as in **`ALTER QLOCAL(ANDREW) TRIGGER NOTRIGGER`** is not valid.

Even though MQSC commands are not case-sensitive, this course represents all MQSC commands in uppercase to distinguish MQSC commands from control commands.



IBM Training

MQSC command summary

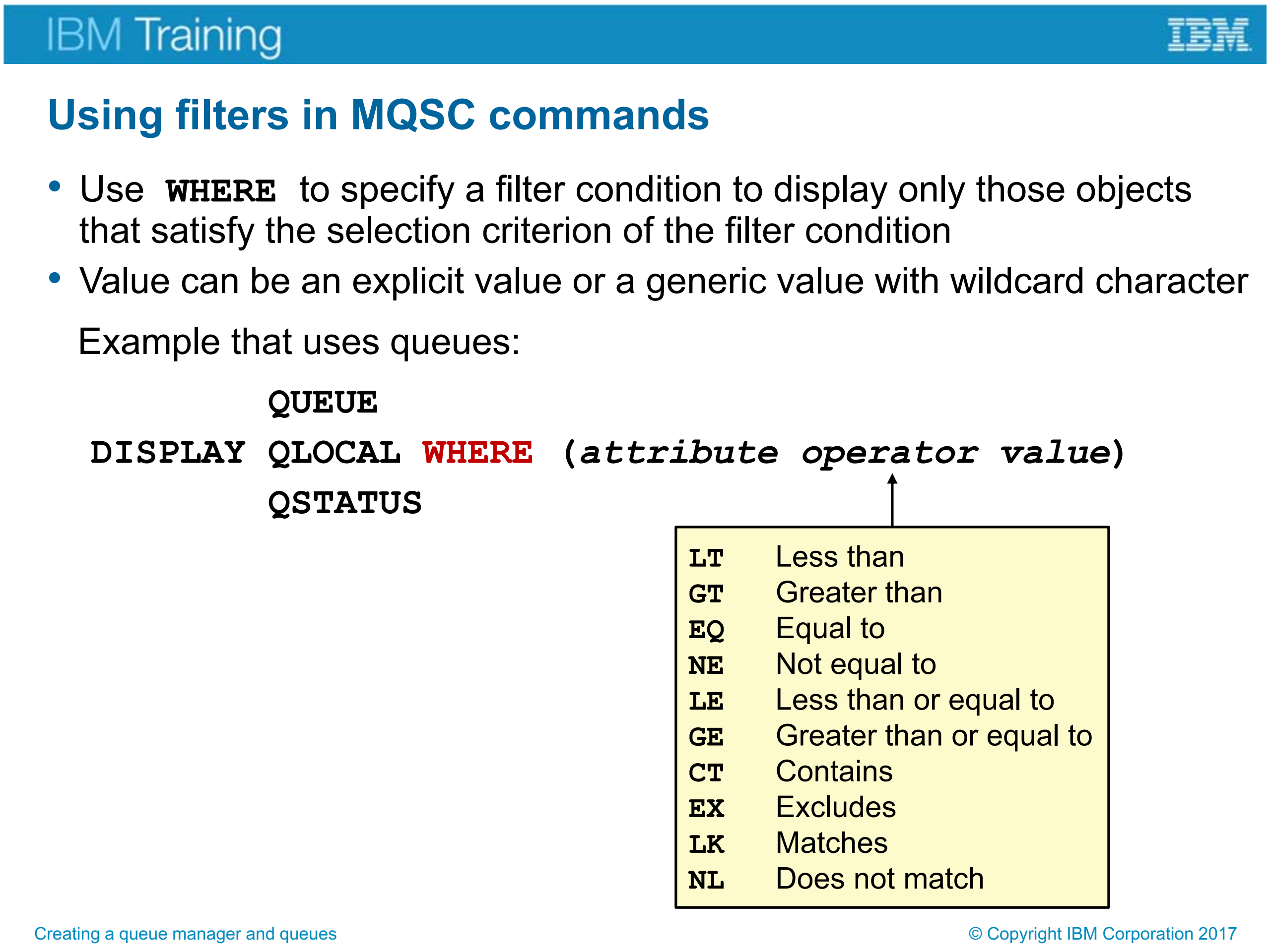
Verbs	MQ objects	Name of the MQ object
ALTER	Queues	• Most objects have specific names
CLEAR	Queue managers	• Wildcard value can be used for displays
DEFINE or DEF	Process definitions	
DELETE or DEL	Namelists	
DISPLAY or DIS	Authentication information	
PING	Channels	
PURGE	Client connection channels	**Examples:**
REFRESH	Listeners	`CREATE QLOCAL(APP.IN) + DEFPSIST(YES)`
RESET	Topic objects	`DIS QL(APP*) CURDEPTH`
RESOLVE		`START CHANNEL(MQ00.MQ03)`
RESUME		`ALTER QMGR(QM00) DEADQ(NEW.DLQ)`
SET		`DIS QSTATUS(BILL.IN) LPUTDATE + LPUTTIME CURDEPTH`
START		
STOP		
SUSPEND		

Creating a queue manager and queues

© Copyright IBM Corporation 2017

Figure 3-18. MQSC command summary

IBM MQ script commands are available to define, change, delete, start (where applicable), display, or do most of the day-to-day administration functions. MQSC commands start with a verb, and are followed by an MQ object name. Several attributes or qualifiers can follow the verb-object name combination.



IBM Training

Using filters in MQSC commands

- Use **WHERE** to specify a filter condition to display only those objects that satisfy the selection criterion of the filter condition
- Value can be an explicit value or a generic value with wildcard character

Example that uses queues:

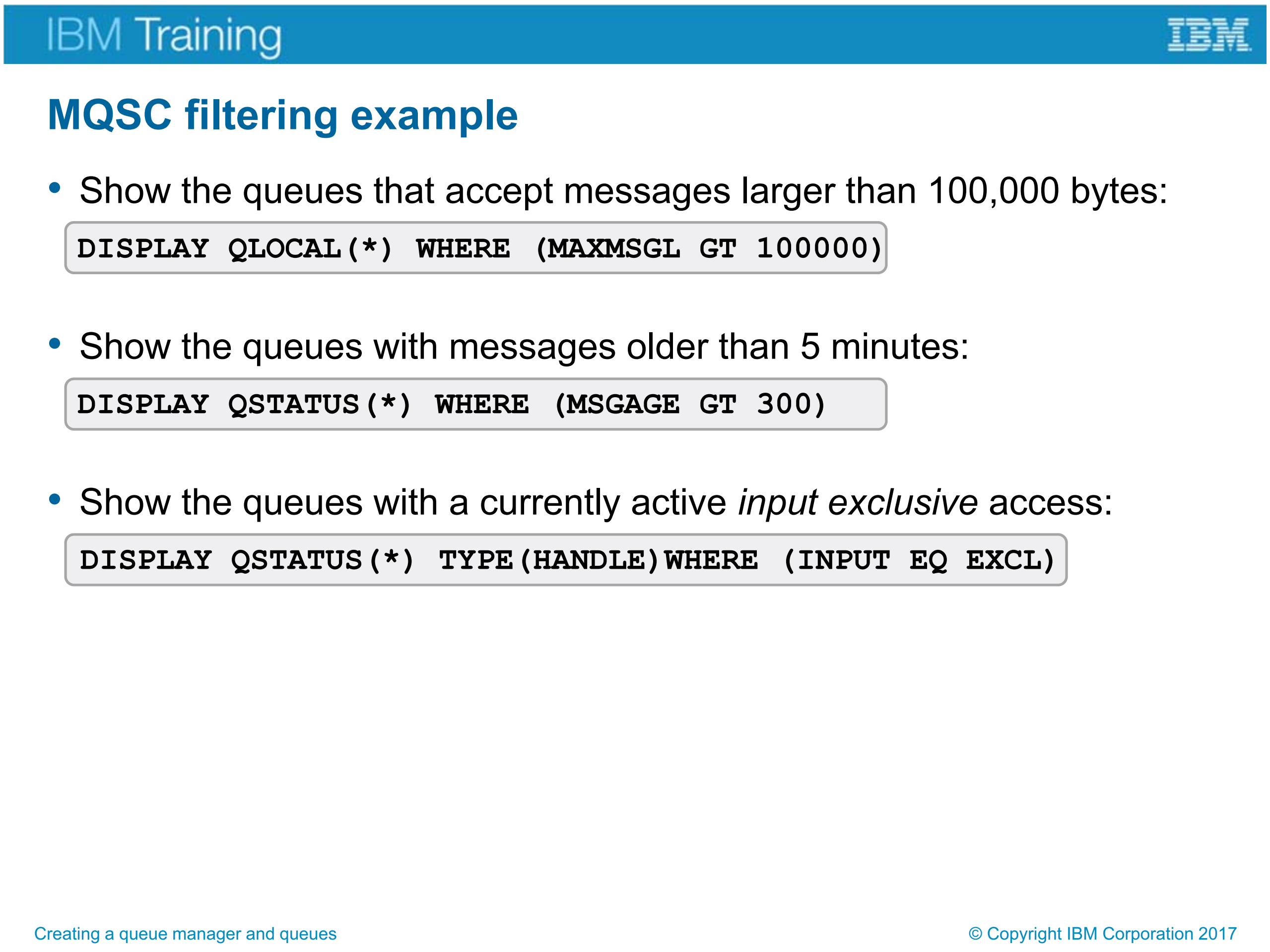
```
        QUEUE
DISPLAY QLOCAL WHERE (attribute operator value)
        QSTATUS
```

LT	Less than
GT	Greater than
EQ	Equal to
NE	Not equal to
LE	Less than or equal to
GE	Greater than or equal to
CT	Contains
EX	Excludes
LK	Matches
NL	Does not match

Creating a queue manager and queues © Copyright IBM Corporation 2017

Figure 3-19. Using filters in MQSC commands

You can use filters on the MQSC **DISPLAY** command to return a subset of data that matches a filter expression.



IBM Training

MQSC filtering example

- Show the queues that accept messages larger than 100,000 bytes:

```
DISPLAY QLOCAL(*) WHERE (MAXMSGL GT 100000)
```

- Show the queues with messages older than 5 minutes:

```
DISPLAY QSTATUS(*) WHERE (MSGAGE GT 300)
```

- Show the queues with a currently active *input exclusive* access:

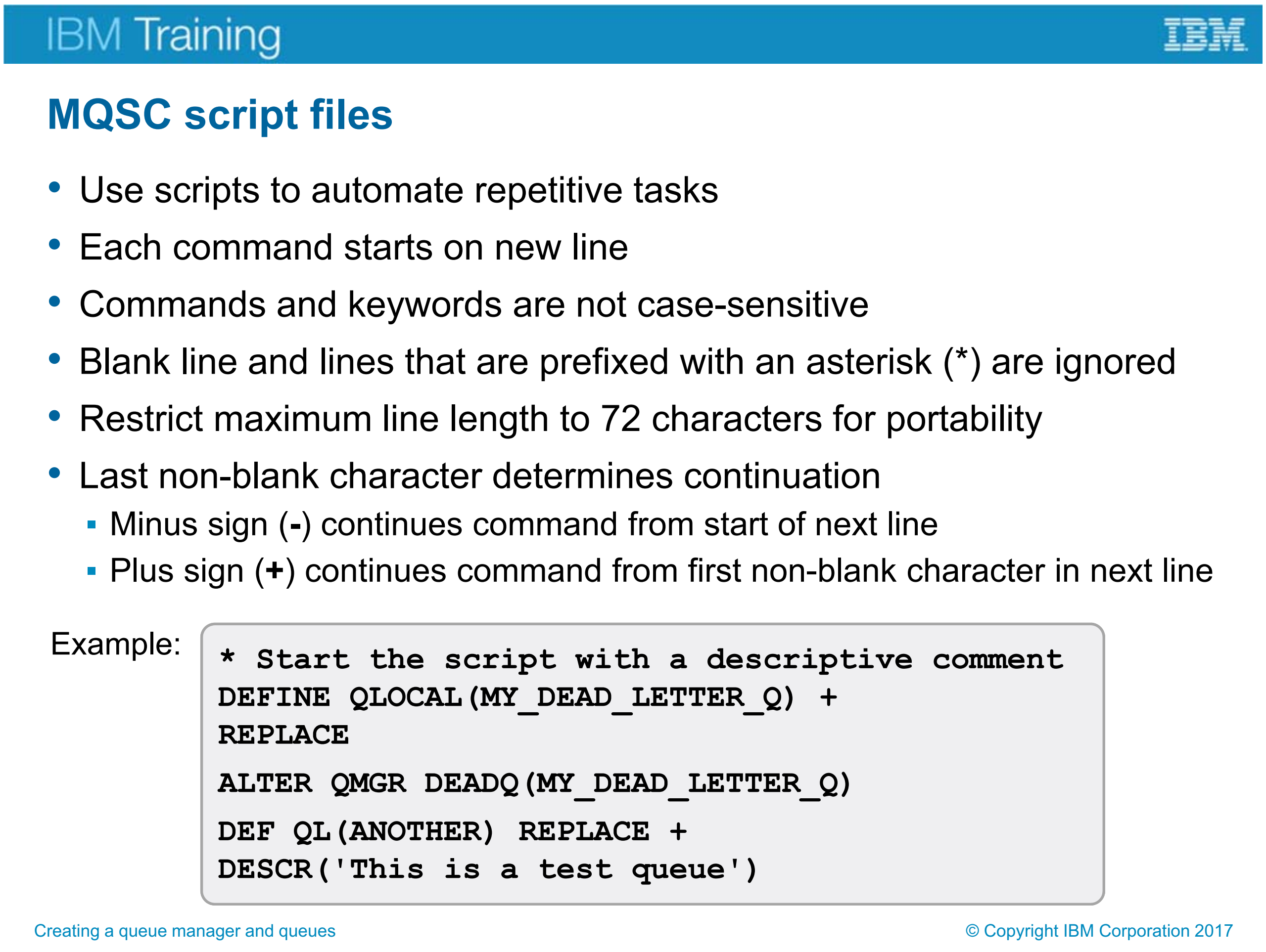
```
DISPLAY QSTATUS(*) TYPE(HANDLE)WHERE (INPUT EQ EXCL)
```

Creating a queue manager and queues

© Copyright IBM Corporation 2017

Figure 3-20. MQSC filtering example

This figure shows examples of MQSC **DISPLAY** commands with filter expressions.



IBM Training

MQSC script files

- Use scripts to automate repetitive tasks
- Each command starts on new line
- Commands and keywords are not case-sensitive
- Blank line and lines that are prefixed with an asterisk (*) are ignored
- Restrict maximum line length to 72 characters for portability
- Last non-blank character determines continuation
  - Minus sign (**-**) continues command from start of next line
  - Plus sign (**+**) continues command from first non-blank character in next line

Example:

```
* Start the script with a descriptive comment
DEFINE QLOCAL(MY_DEAD_LETTER_Q) +
REPLACE
ALTER QMGR DEADQ(MY_DEAD_LETTER_Q)
DEF QL(ANOTHER) REPLACE +
DESCR('This is a test queue')
```

Creating a queue manager and queues © Copyright IBM Corporation 2017

Figure 3-21. MQSC script files

In most cases, administrators create MQSC scripts to complete repetitive administrative functions. MQSC commands can be included in a text file and then run against a queue manager as a script.

This figure lists some of the rules for writing MQSC script files. More syntax rules for writing MQ commands include the following rules:

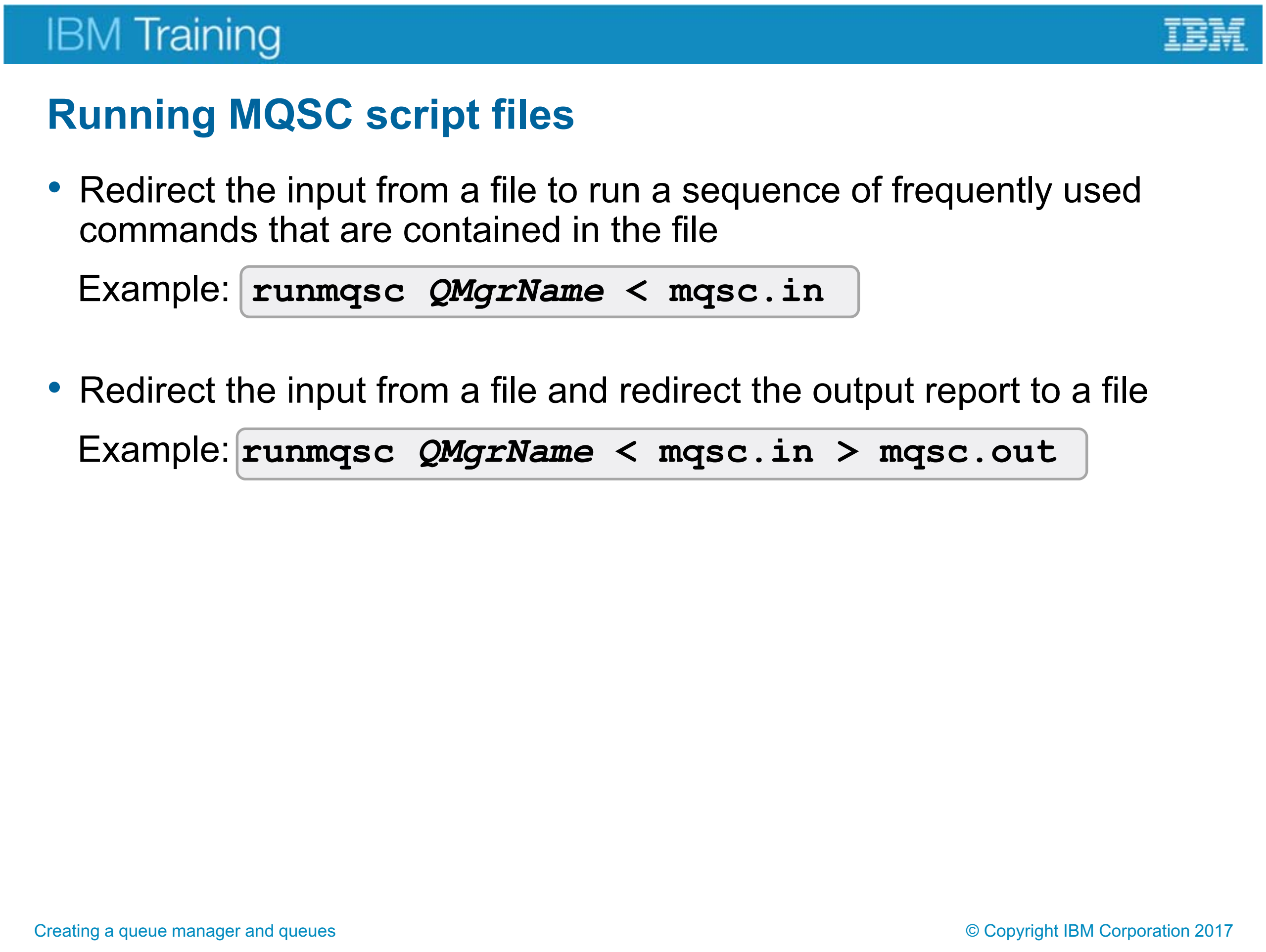
- An MQSC command can contain a string of characters. A string that contains blanks, lowercase characters, or special characters other than characters valid in the name of an MQ object, must be enclosed in single quotation marks. Lowercase characters that are not enclosed in single quotation marks are internally converted to uppercase.
- You can optionally use a semicolon (`;`) to end a command.
- A string that has no characters is not valid.
- The line length is limited to 72 characters. A plus sign (**+**) is a continuation character.

The figure shows three examples of MQSC commands.

- The first command creates a local queue with the name MY_DEAD_LETTER_Q.
- The second command alters the queue manager by declaring MY_DEAD_LETTER_Q as the dead-letter queue of the queue manager.

- The third command creates another local queue. A keyword, such as `DESCR`, can be in lowercase or uppercase. In this course, all MQSC commands and keywords are shown in uppercase to distinguish them from control commands.

  The single quotation marks that enclose the string `This is a test` queue are required because the string contains blanks.



IBM Training

Running MQSC script files

- Redirect the input from a file to run a sequence of frequently used commands that are contained in the file

  Example:
  ```
  runmqsc QMgrName < mqsc.in
  ```

- Redirect the input from a file and redirect the output report to a file

  Example:
  ```
  runmqsc QMgrName < mqsc.in > mqsc.out
  ```

Creating a queue manager and queues

© Copyright IBM Corporation 2017

Figure 3-22. Running MQSC script files

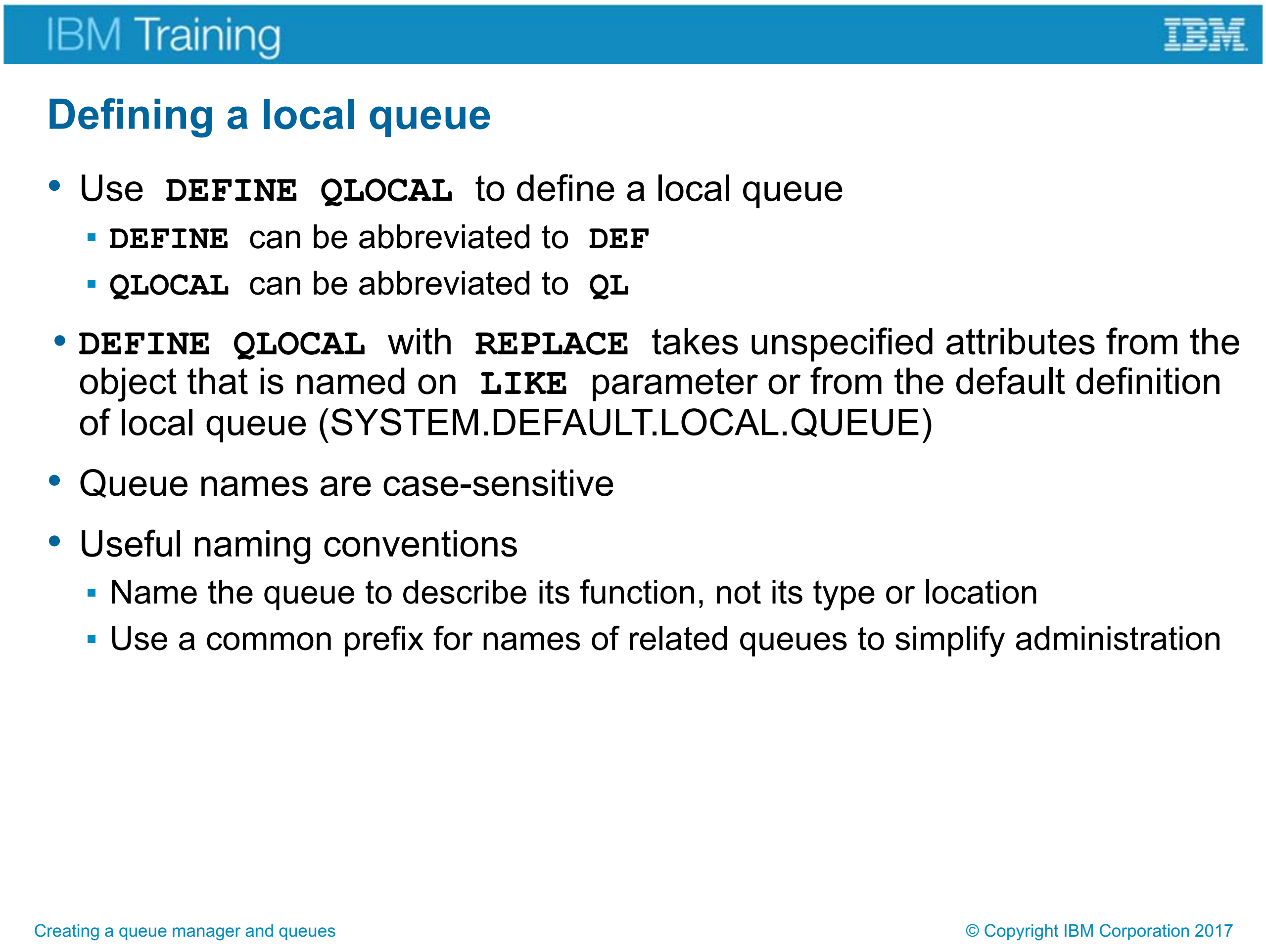
As shown in the figure, the **runmqsc** command can be used to process scripts. It reads from the *standard input device* and writes to the *standard output device*. Typically, the standard input device is the keyboard and the standard output device is the display. However, by using redirection operators, input can be taken from a file and output can be directed to a file.

In the first example, the queue manager that is identified with the **runmqsc** command reads and processes the file that is named **mqsc.in**.

In the second example, the queue manager that is identified with the **runmqsc** command reads and processes the file that is named **mqsc.in**. A report that shows the processing of each command is created in the **mqsc.out** file.

It is useful to maintain MQSC files, for the following reasons:

- To replicate a queue manager configuration on multiple systems
- To recover a queue manager configuration
- When you go through a number of iterations in testing a queue manager configuration



IBM Training

Defining a local queue

- Use **`DEFINE QLOCAL`** to define a local queue
  - **`DEFINE`** can be abbreviated to **`DEF`**
  - **`QLOCAL`** can be abbreviated to **`QL`**
- **`DEFINE QLOCAL`** with **`REPLACE`** takes unspecified attributes from the object that is named on **`LIKE`** parameter or from the default definition of local queue (SYSTEM.DEFAULT.LOCAL.QUEUE)
- Queue names are case-sensitive
- Useful naming conventions
  - Name the queue to describe its function, not its type or location
  - Use a common prefix for names of related queues to simplify administration

Creating a queue manager and queues © Copyright IBM Corporation 2017

Figure 3-23. Defining a local queue

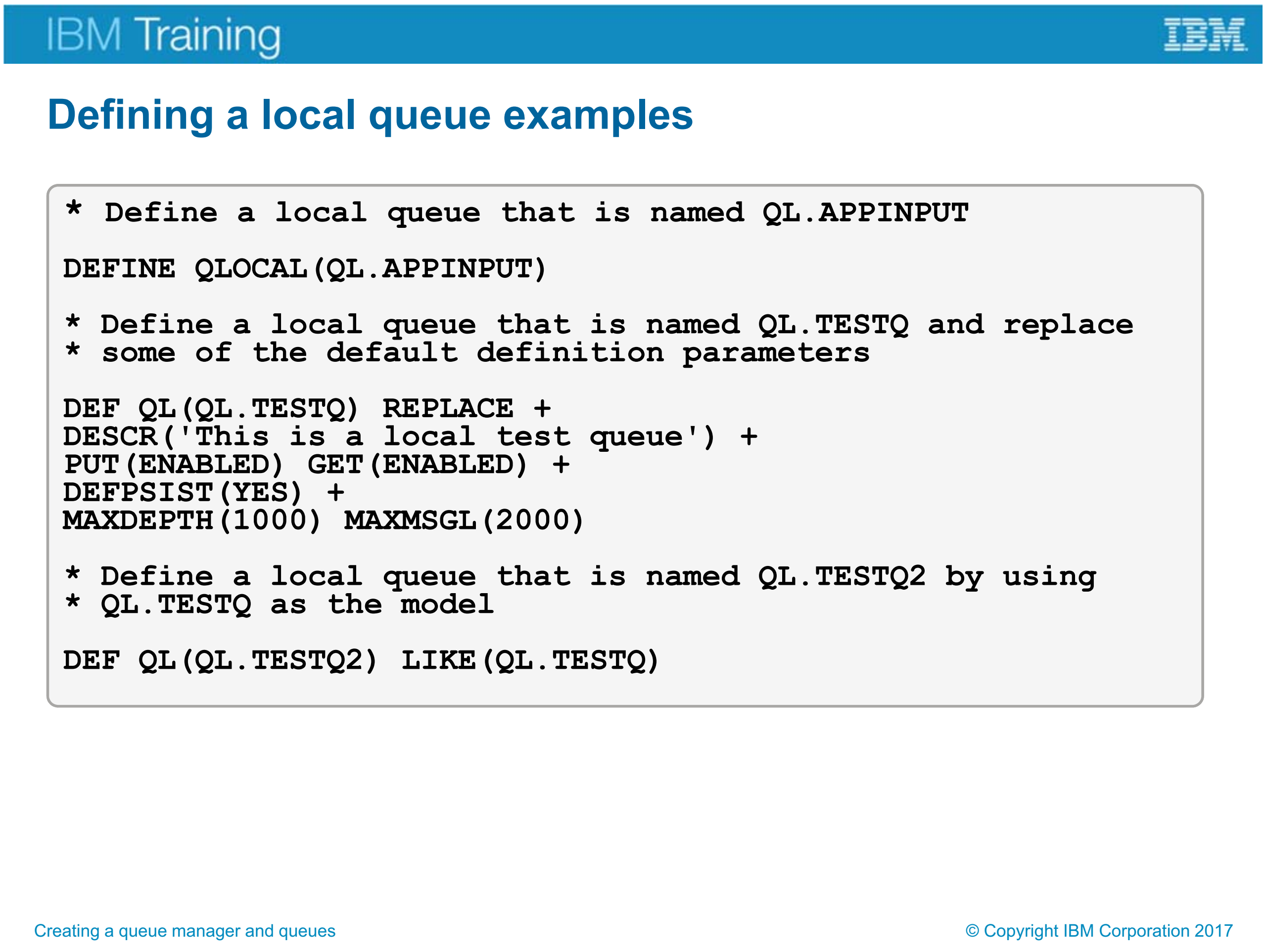
The MQSC command **`DEFINE QLOCAL`** or its synonym of **`DEF QL`** defines a queue that is local to the queue manager.

Keywords and their values specify the values of attributes of the local queue that is created. The values of the attributes that are not explicitly defined are taken from the values of the corresponding attributes of the default local queue, SYSTEM.DEFAULT.LOCAL.QUEUE. The SYSTEM.DEFAULT.LOCAL.QUEUE is created during MQ installation.

Every queue is owned by a queue manager and possesses a name. An application identifies a queue by its name. Queue names in MQ are case-sensitive. Use a standard convention for case such as all uppercase or all lowercase.

Other queue name guidelines include the following rules:

- Do not use the type or location of the queue in the queue name. If a queue is changed from a local to a remote queue, for example, the same name can still be used for the queue; it is not necessary to change the applications that reference the queue. Instead, name the queue after its function.
- Use a common prefix for the names of related queues to aid administration. For example, name all queues that are related to the same application with the same prefix.



IBM Training

Defining a local queue examples

```
* Define a local queue that is named QL.APPINPUT

DEFINE QLOCAL(QL.APPINPUT)

* Define a local queue that is named QL.TESTQ and replace
* some of the default definition parameters

DEF QL(QL.TESTQ) REPLACE +
DESCR('This is a local test queue') +
PUT(ENABLED) GET(ENABLED) +
DEFPSIST(YES) +
MAXDEPTH(1000) MAXMSGL(2000)

* Define a local queue that is named QL.TESTQ2 by using
* QL.TESTQ as the model

DEF QL(QL.TESTQ2) LIKE(QL.TESTQ)
```

Creating a queue manager and queues © Copyright IBM Corporation 2017

Figure 3-24. Defining a local queue examples

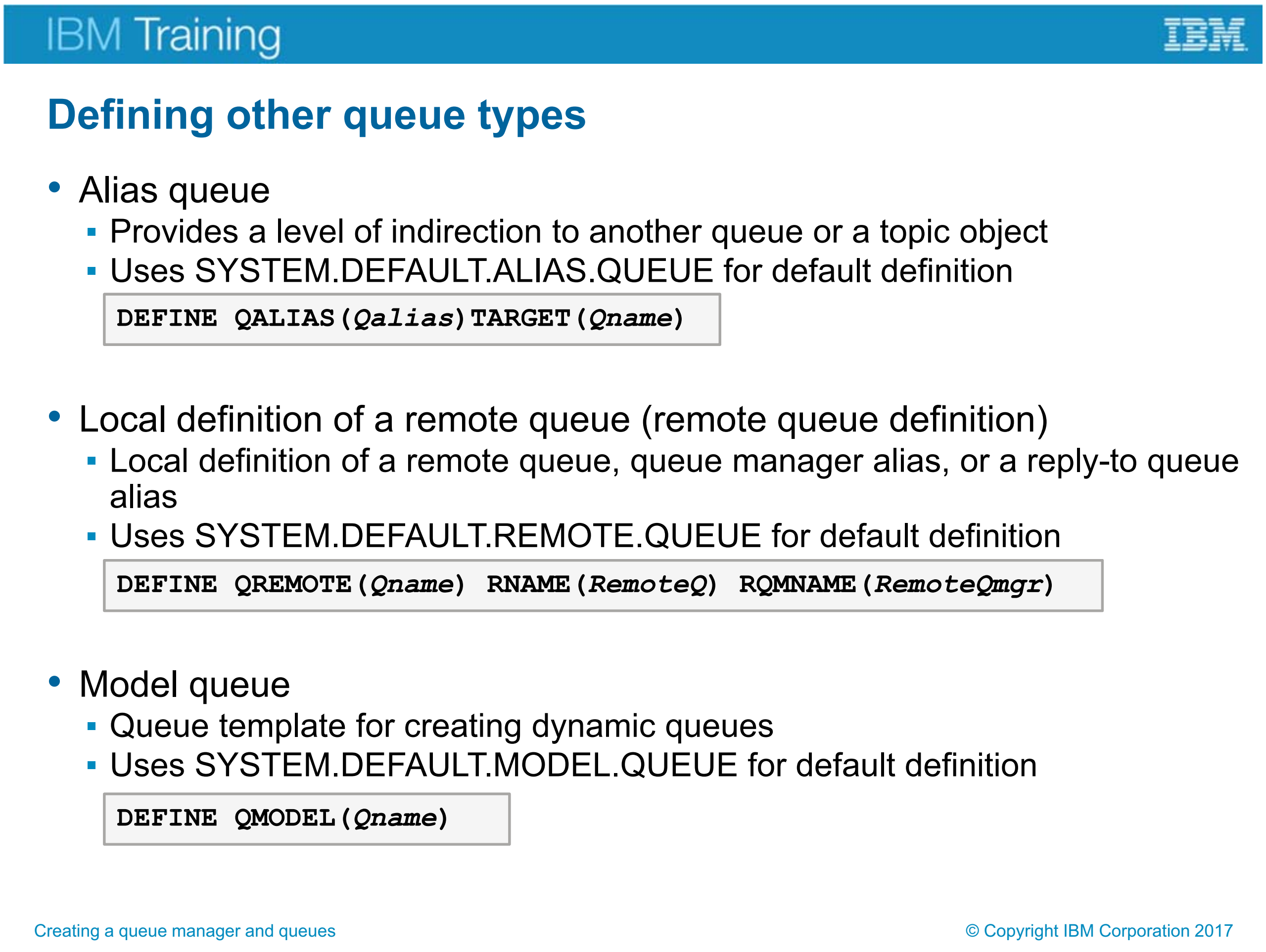
The figure shows three examples of the MQSC command for defining a local queue. The second and third examples use the synonym **DEF QL**.

The first example creates a local queue that is named QL.APPINPUT and uses the default attributes of SYSTEM.DEFAULT.LOCAL.QUEUE for a local queue.

The second example creates a local queue that is named QL.TESTQ and replaces some of the default attributes. The **REPLACE** keyword indicates that if the queue exists, replace its definition with the new one as defined in this command. Any messages on an existing queue are retained.

The third example defines a local queue that is named QL.TEST2. The **LIKE** keyword means that the named queue (QL.TESTQ in this example) is used for the default values of attributes rather than the default local queue SYSTEM.DEFAULT.LOCAL.QUEUE.

This unit provides an overview of command syntax. You learn more about creating queues and queue attributes throughout this course and in the lab exercises.


IBM Training

Defining other queue types

- Alias queue
  - Provides a level of indirection to another queue or a topic object
  - Uses SYSTEM.DEFAULT.ALIAS.QUEUE for default definition

```
DEFINE QALIAS(Qalias)TARGET(Qname)
```

- Local definition of a remote queue (remote queue definition)
  - Local definition of a remote queue, queue manager alias, or a reply-to queue alias
  - Uses SYSTEM.DEFAULT.REMOTE.QUEUE for default definition

```
DEFINE QREMOTE(Qname) RNAME(RemoteQ) RQMNAME(RemoteQmgr)
```

- Model queue
  - Queue template for creating dynamic queues
  - Uses SYSTEM.DEFAULT.MODEL.QUEUE for default definition

```
DEFINE QMODEL(Qname)
```

Creating a queue manager and queues © Copyright IBM Corporation 2017

Figure 3-25. Defining other queue types

MQ supports other queue types.

An *alias queue* is an MQ object that refers indirectly to another queue. The MQSC command to create an alias queue is **DEFINE QALIAS** or **DEF QA**. The **TARGET** keyword specifies the name of the queue to which the alias queue resolves. The target queue can be a local queue or a local definition of a remote queue.

A *local definition of a remote queue*, or a *remote queue definition*, is an MQ object owned by one queue manager that refers to a queue owned by another queue manager. The MQSC command to create a local definition of a remote queue is **DEFINE QREMOTE** or **DEF QR**. The keyword **RNAME** is the name of the queue as it is known on the remote queue manager. The keyword **RQMNAME** is the name of the remote queue manager.

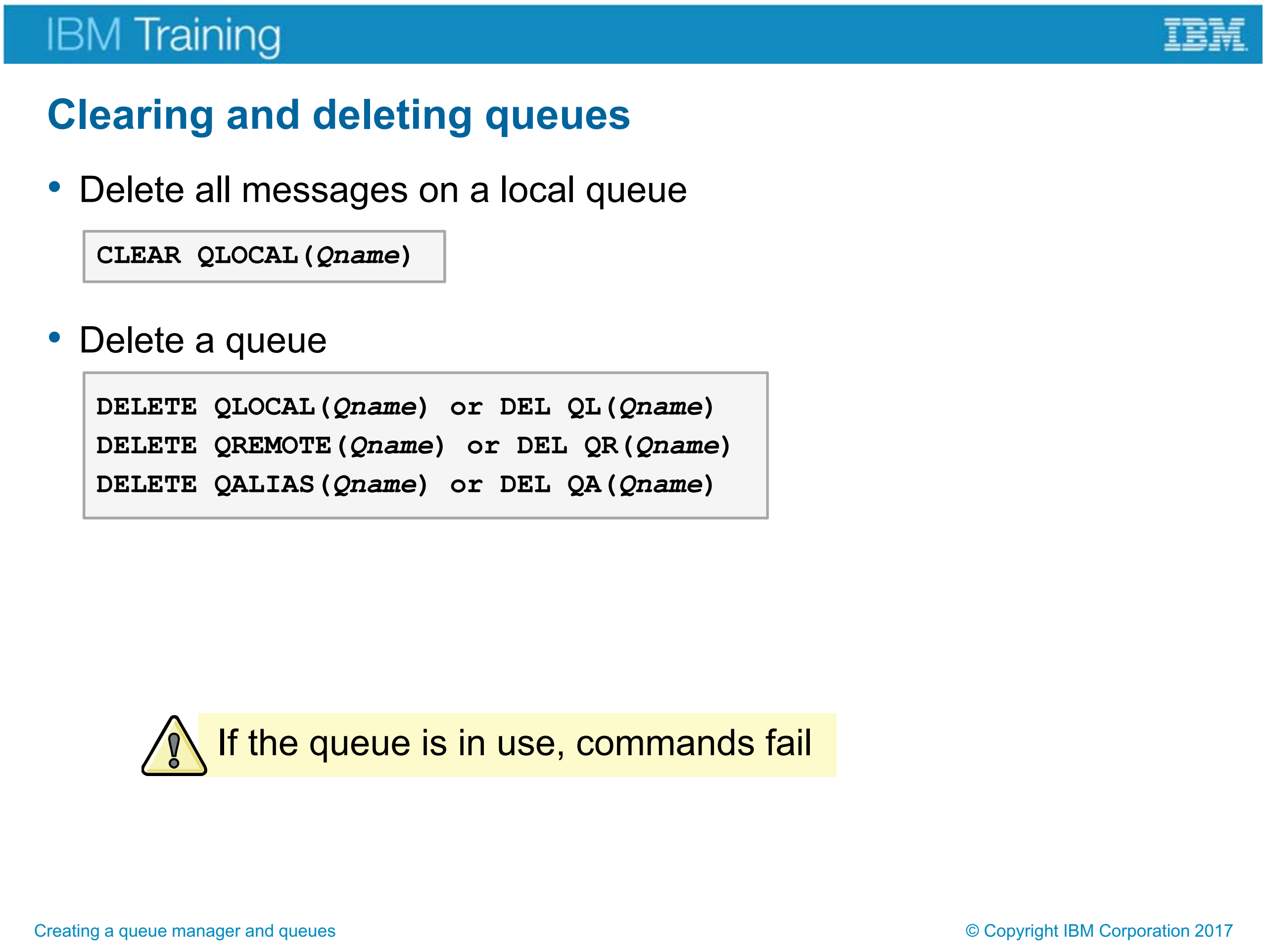
A *model queue* is an MQ object whose attributes are used as a template for creating a dynamic queue. The MQSC command to create a model queue is **DEFINE QMODEL**. When an application opens a model queue, the queue manager creates a dynamic queue. The **DEFTYPE** keyword specifies whether a dynamic queue that is created from the model queue is one of the following types:

- A *temporary* dynamic queue, which is deleted when it is closed and does not survive a queue manager restart

- A *permanent* dynamic queue, whose deletion on the MQCLOSE call is optional and which does survive a queue manager restart.

Of the two types of dynamic queues, only a permanent dynamic queue can store persistent messages. A typical use of a dynamic queue is as a reply-to queue for a client program that is sending requests to a server.

Of the four types of queues, only a local queue can store messages.



IBM Training

Clearing and deleting queues

- Delete all messages on a local queue

```
CLEAR QLOCAL(Qname)
```

- Delete a queue

```
DELETE QLOCAL(Qname) or DEL QL(Qname)
DELETE QREMOTE(Qname) or DEL QR(Qname)
DELETE QALIAS(Qname) or DEL QA(Qname)
```

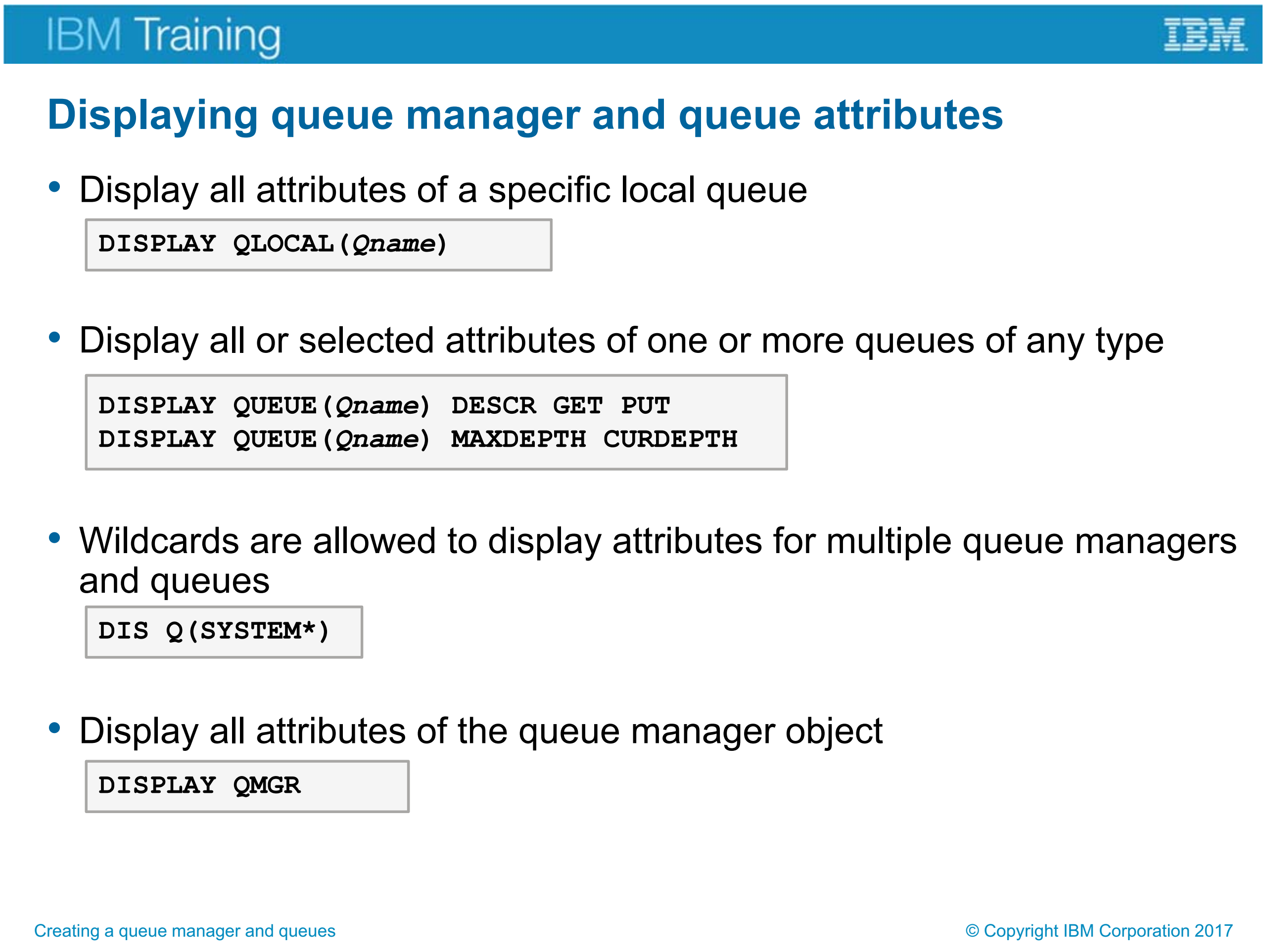
If the queue is in use, commands fail

Creating a queue manager and queues © Copyright IBM Corporation 2017

Figure 3-26. Clearing and deleting queues

This figure shows the MQSC commands that can be used to clear all messages on a queue and delete a queue. These commands fail if the queue:

- Contains uncommitted messages that are put on the queue under sync point
- Is open by an application (with any open options)
- Is a transmission queue, and any queue that is, or resolves to, a remote queue that references this transmission queue, is open



IBM Training

Displaying queue manager and queue attributes

- Display all attributes of a specific local queue

```
DISPLAY QLOCAL(Qname)
```

- Display all or selected attributes of one or more queues of any type

```
DISPLAY QUEUE(Qname) DESCR GET PUT
DISPLAY QUEUE(Qname) MAXDEPTH CURDEPTH
```

- Wildcards are allowed to display attributes for multiple queue managers and queues

```
DIS Q(SYSTEM*)
```

- Display all attributes of the queue manager object

```
DISPLAY QMGR
```

Creating a queue manager and queues © Copyright IBM Corporation 2017

Figure 3-27. Displaying queue manager and queue attributes

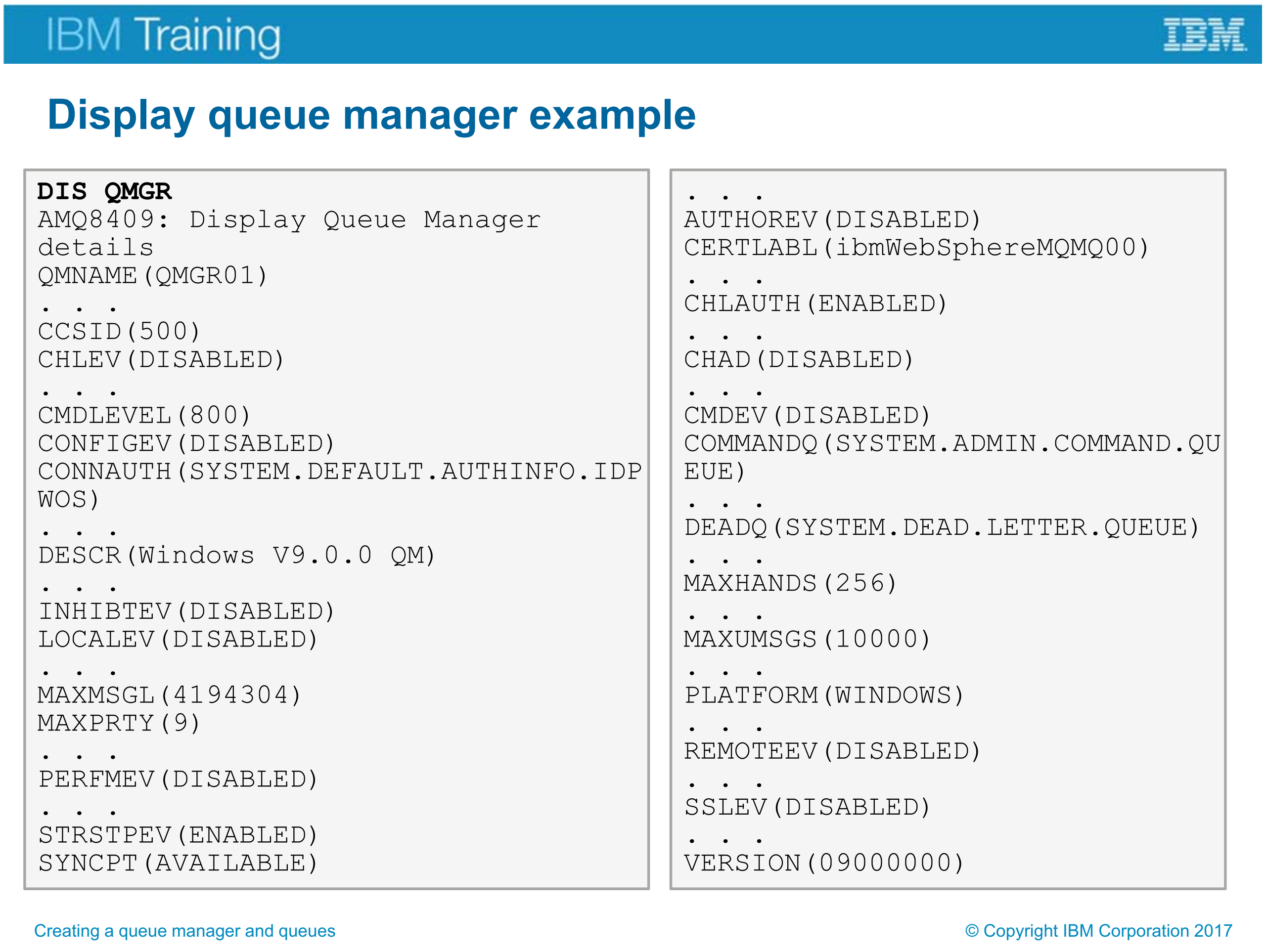
The MQSC command to display all the attributes of a specific queue is **DISPLAY** or its synonym **DIS** followed by the queue type keyword (**QLOCAL**, **QALIAS**, **QREMOTE**, or **QMODEL**).

To display selected attributes for one or more queues, use the **DISPLAY QUEUE** command. This command applies to all types of queue: local, alias, remote, and model. The **DISPLAY QUEUE** command can specify a generic queue name by using a wildcard trailing asterisk (*). An asterisk without any other characters specifies “all queues”.

The default behavior of the **DISPLAY QUEUE** command is to display all queue attributes. You can request the display of selected attributes by using a **WHERE** clause.

The MQSC command to display the attributes of the queue manager object is **DISPLAY QMGR**. Do not enter the name of the queue manager in the command.

The default action of the **DISPLAY QMGR** command is to display all the attributes. You can choose to display different sets of queue manager parameters, such as **CHINIT**, **CLUSTER**, **EVENT**, **SYSTEM**, and **PUBSUB**.



IBM Training

Display queue manager example

```
DIS QMGR
AMQ8409: Display Queue Manager
details
QMNAME(QMGR01)
. . .
CCSID(500)
CHLEV(DISABLED)
. . .
CMDLEVEL(800)
CONFIGEV(DISABLED)
CONNAUTH(SYSTEM.DEFAULT.AUTHINFO.IDP
WOS)
. . .
DESCR(Windows V9.0.0 QM)
. . .
INHIBTEV(DISABLED)
LOCALEV(DISABLED)
. . .
MAXMSGL(4194304)
MAXPRTY(9)
. . .
PERFMEV(DISABLED)
. . .
STRSTPEV(ENABLED)
SYNCPT(AVAILABLE)
```

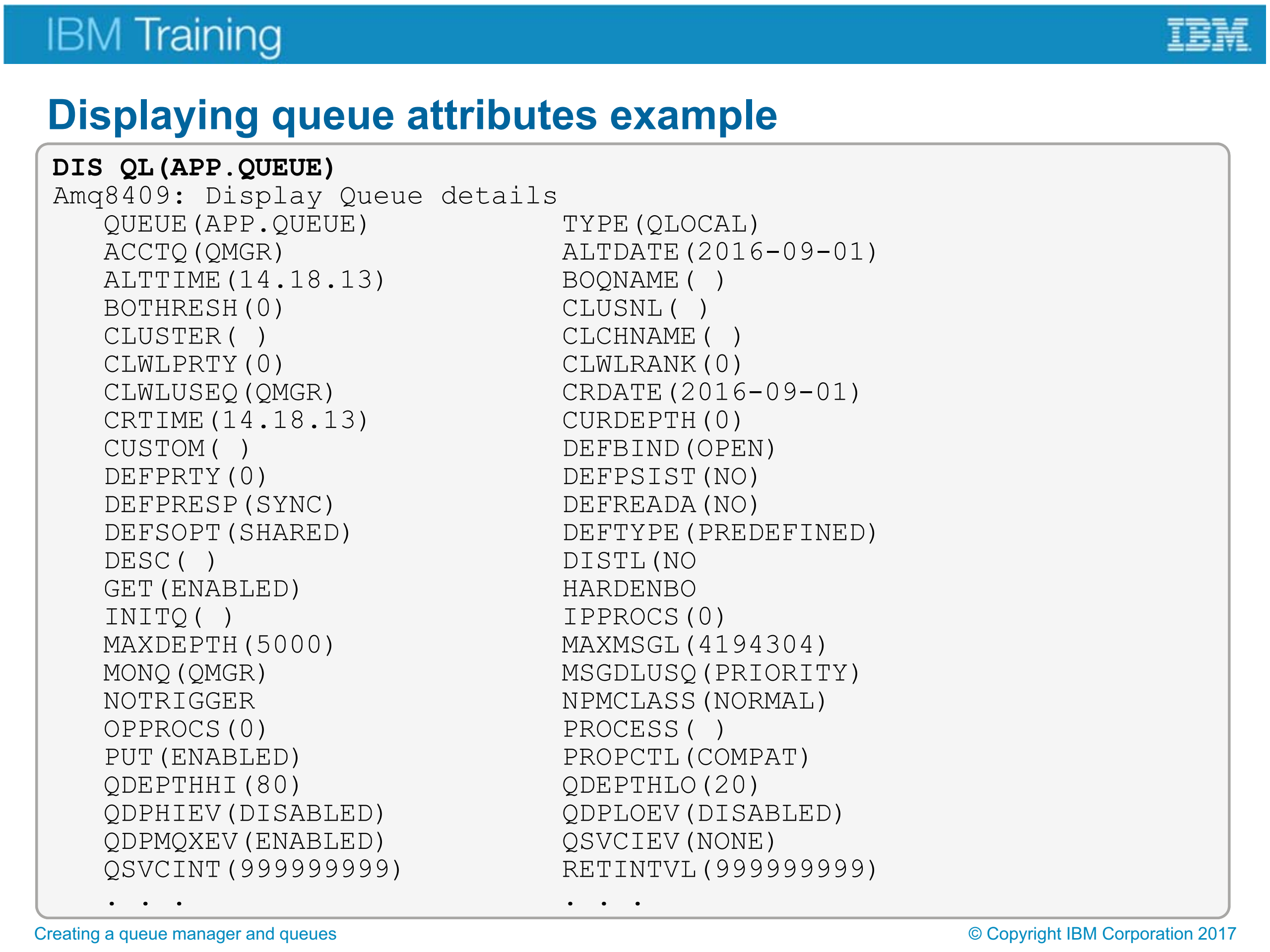
```
. . .
AUTHOREV(DISABLED)
CERTLABL(ibmWebSphereMQMQ00)
. . .
CHLAUTH(ENABLED)
. . .
CHAD(DISABLED)
. . .
CMDEV(DISABLED)
COMMANDQ(SYSTEM.ADMIN.COMMAND.QU
EUE)
. . .
DEADQ(SYSTEM.DEAD.LETTER.QUEUE)
. . .
MAXHANDS(256)
. . .
MAXUMSGS(10000)
. . .
PLATFORM(WINDOWS)
. . .
REMOTEEV(DISABLED)
. . .
SSLEV(DISABLED)
. . .
VERSION(09000000)
```

Creating a queue manager and queues

© Copyright IBM Corporation 2017

Figure 3-28. Display queue manager example

This figure shows an example of the results from entering the **`DIS QMGR`** command. This example shows a subset of the results only.



IBM Training

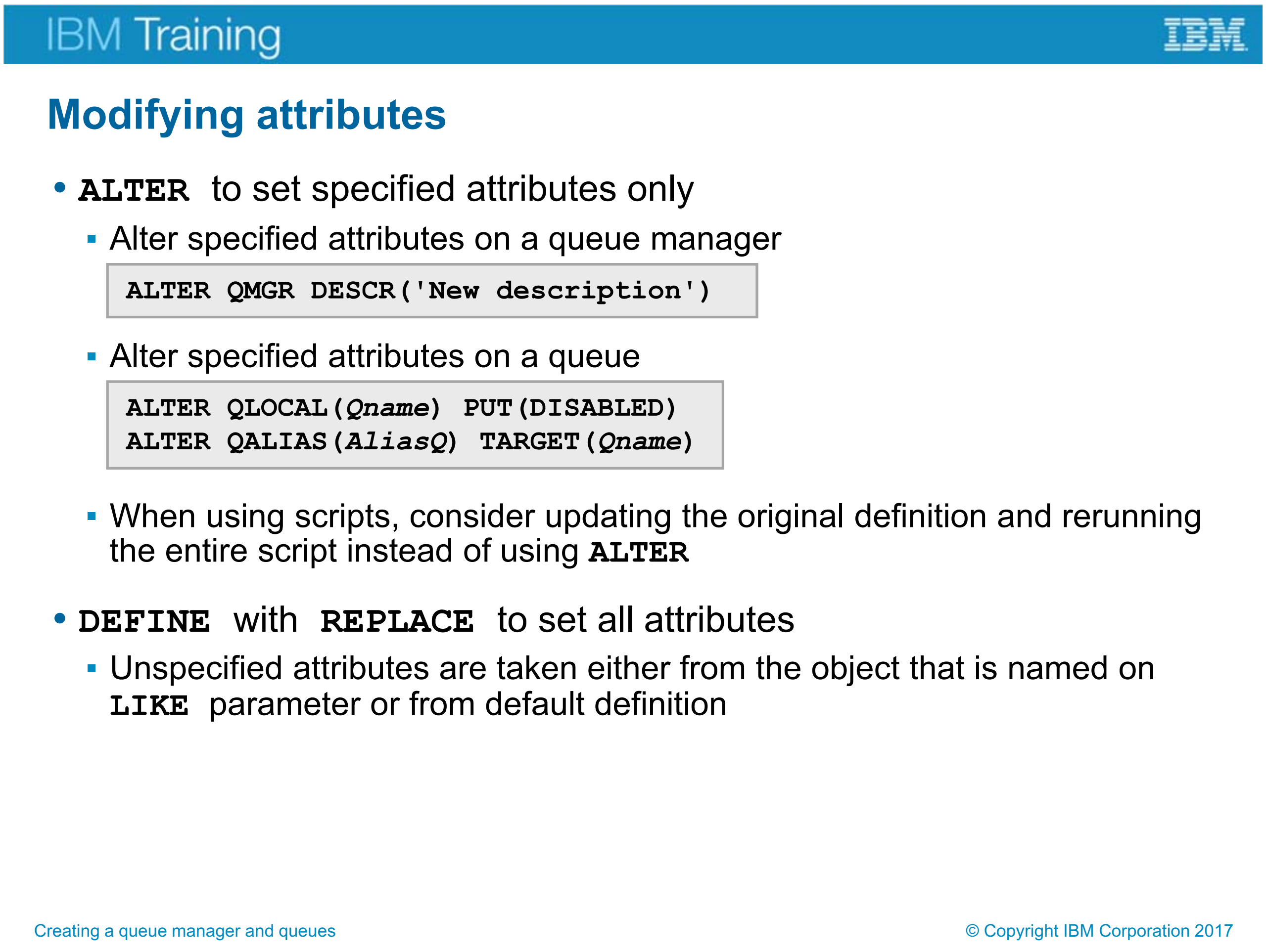
Displaying queue attributes example

```
DIS QL(APP.QUEUE)
Amq8409: Display Queue details
   QUEUE(APP.QUEUE)                     TYPE(QLOCAL)
   ACCTQ(QMGR)                          ALTDATE(2016-09-01)
   ALTTIME(14.18.13)                    BOQNAME( )
   BOTHRESH(0)                          CLUSNL( )
   CLUSTER( )                           CLCHNAME( )
   CLWLPRTY(0)                          CLWLRANK(0)
   CLWLUSEQ(QMGR)                       CRDATE(2016-09-01)
   CRTIME(14.18.13)                     CURDEPTH(0)
   CUSTOM( )                            DEFBIND(OPEN)
   DEFPRTY(0)                           DEFPSIST(NO)
   DEFPRESP(SYNC)                       DEFREADA(NO)
   DEFSOPT(SHARED)                      DEFTYPE(PREDEFINED)
   DESC( )                              DISTL(NO
   GET(ENABLED)                         HARDENBO
   INITQ( )                             IPPROCS(0)
   MAXDEPTH(5000)                       MAXMSGL(4194304)
   MONQ(QMGR)                           MSGDLUSQ(PRIORITY)
   NOTRIGGER                            NPMCLASS(NORMAL)
   OPPROCS(0)                           PROCESS( )
   PUT(ENABLED)                         PROPCTL(COMPAT)
   QDEPTHHI(80)                         QDEPTHLO(20)
   QDPHIEV(DISABLED)                    QDPLOEV(DISABLED)
   QDPMQXEV(ENABLED)                    QSVCIEV(NONE)
   QSVCINT(999999999)                   RETINTVL(999999999)
   . . .                                . . .
```

Creating a queue manager and queues © Copyright IBM Corporation 2017

Figure 3-29. Displaying queue attributes example

This figure shows an example of the results that are returned from the **DISPLAY QLOCAL** (**DIS QL**) command.



IBM Training

Modifying attributes

- **`ALTER`** to set specified attributes only
  - Alter specified attributes on a queue manager

    ```
    ALTER QMGR DESCR('New description')
    ```

  - Alter specified attributes on a queue

    ```
    ALTER QLOCAL(Qname) PUT(DISABLED)
    ALTER QALIAS(AliasQ) TARGET(Qname)
    ```

  - When using scripts, consider updating the original definition and rerunning the entire script instead of using **`ALTER`**
- **`DEFINE`** with **`REPLACE`** to set all attributes
  - Unspecified attributes are taken either from the object that is named on **`LIKE`** parameter or from default definition

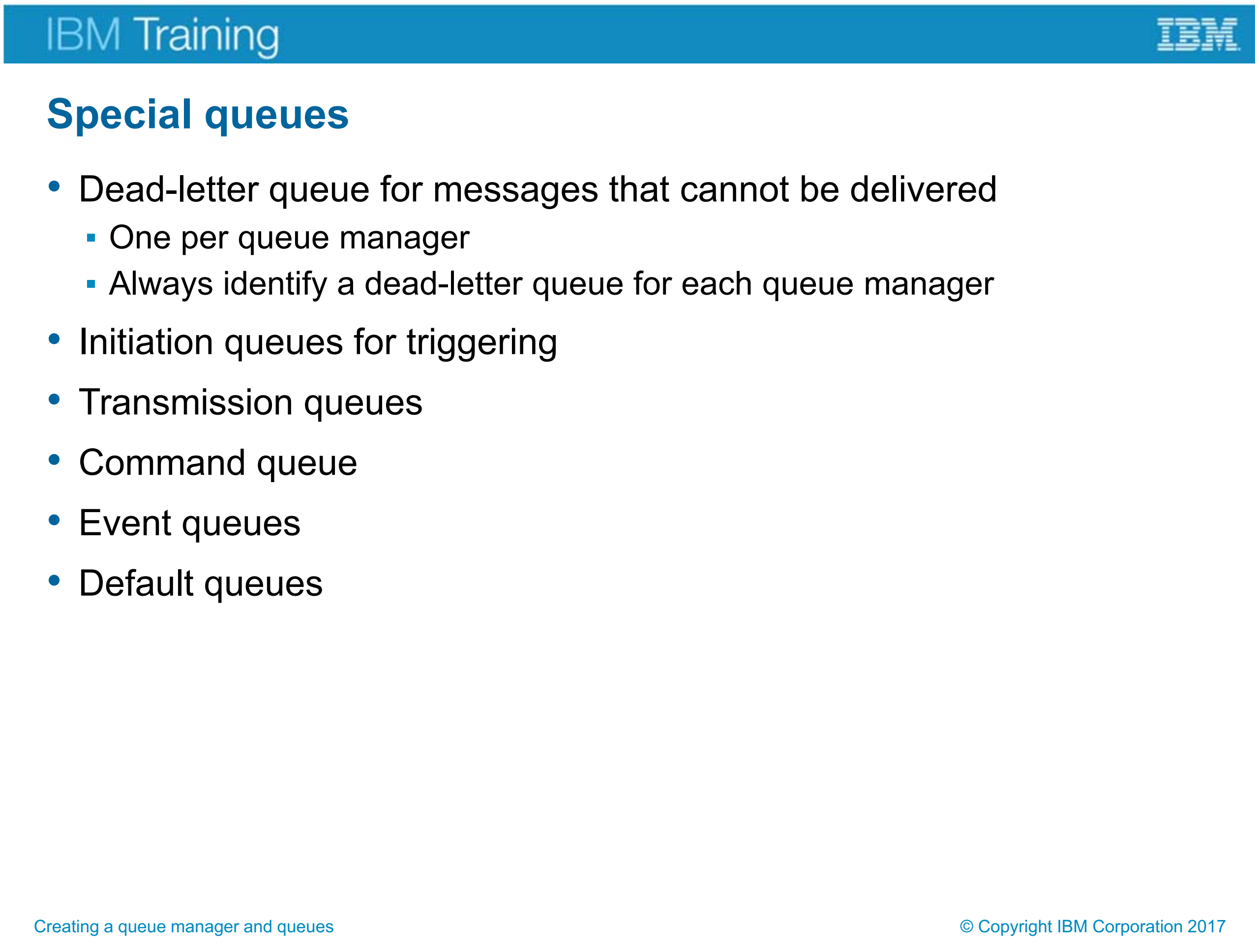
Creating a queue manager and queues © Copyright IBM Corporation 2017

Figure 3-30. Modifying attributes

The MQSC command **ALTER QMGR** is used to modify queue manager attributes.

The MQSC commands **ALTER QLOCAL**, **ALTER QALIAS**, **ALTER QREMOTE**, and **ALTER QMODEL** change the specified attributes of the queue to which the command is applied. The remaining attributes of the queue are left unchanged.

If you attempt to alter a queue and the queue is in use, the command fails.



IBM Training

Special queues

- Dead-letter queue for messages that cannot be delivered
  - One per queue manager
  - Always identify a dead-letter queue for each queue manager
- Initiation queues for triggering
- Transmission queues
- Command queue
- Event queues
- Default queues

Creating a queue manager and queues © Copyright IBM Corporation 2017

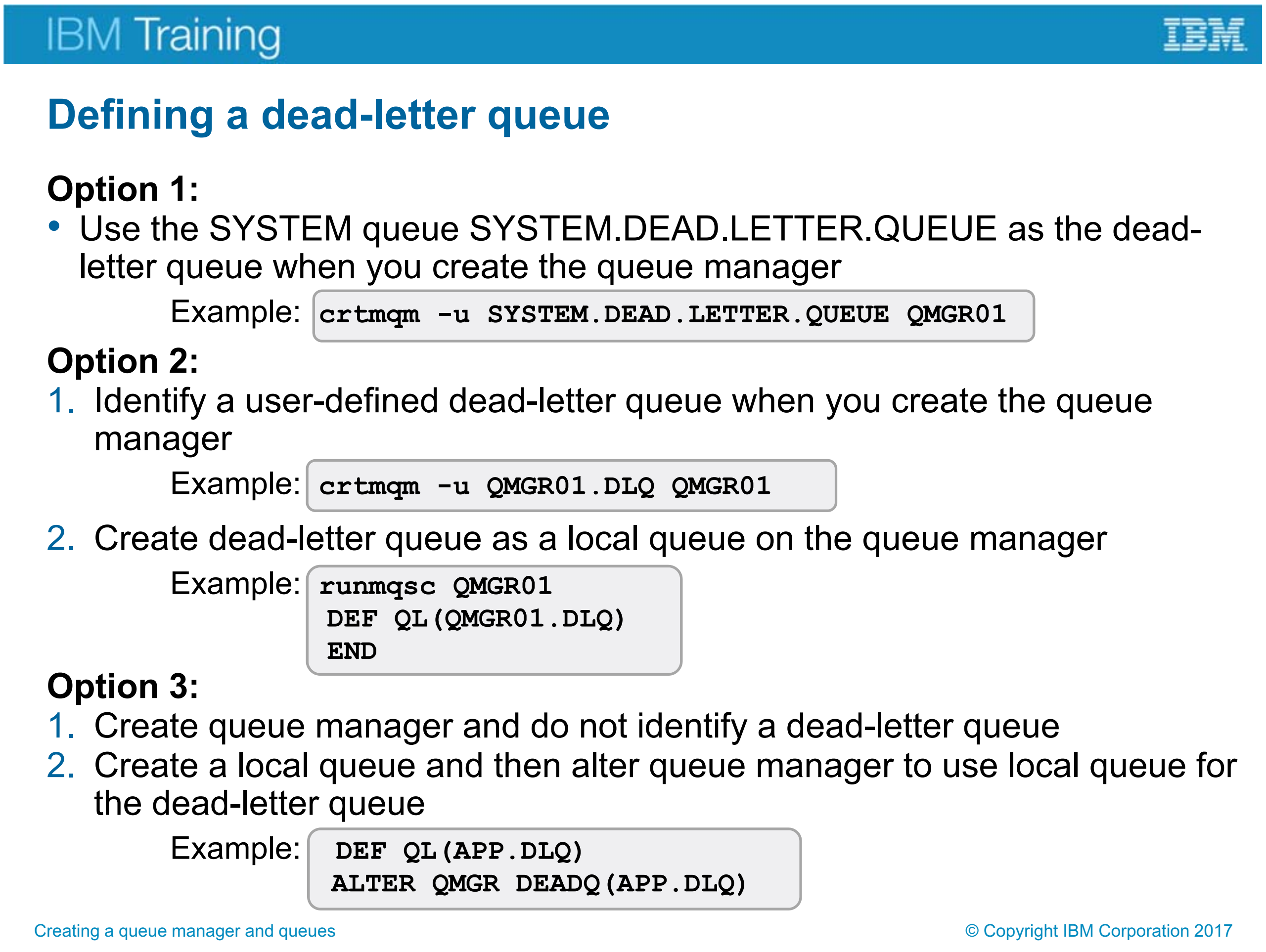
Figure 3-31. Special queues

Some local queues have special purposes in MQ. You learn more about these special-purpose queues throughout this course.

- A *dead-letter queue* is a designated queue where a queue manager puts messages that cannot otherwise be delivered. It is not mandatory for a queue manager to have a dead-letter queue, but it is a good practice and critical to the health of the queue manager. The SYSTEM queue SYSTEM.DEAD.LETTER.QUEUE is not automatically assigned as the queue manager’s dead-letter queue but can be assigned as the dead-letter when the queue manager is created.
- An *initiation queue* is a queue that is used to implement triggering.
- *Transmission queues* are queues that temporarily store messages that are destined for remote queue managers.
- The *command queue*, SYSTEM.ADMIN.COMMAND.QUEUE, is a local queue to which suitably authorized applications can send MQSC commands for processing. The MQ command server retrieves these commands. The command server validates the commands, passes the valid ones on for processing by the queue manager, and returns any responses to the appropriate reply-to queue.
- When a queue manager detects an instrumentation event, which can be either a channel, queue manager, performance, or logger event, the queue manager puts a message that

describes that event on an *event queue*. A system management application can monitor an event queue, get the messages put on these queues, and then take appropriate action.

- The purpose of the *default queues* is to identify the default values of the attributes of any new queue that you create. One default queue exists for each of the four types of queues: local, alias, remote, and model. You need to include in the definition of a queue only those attributes whose values are different from the default values when you create a queue. You can change the default value of an attribute by redefining the appropriate default queue.



IBM Training

Defining a dead-letter queue

Option 1:

- Use the SYSTEM queue SYSTEM.DEAD.LETTER.QUEUE as the dead-letter queue when you create the queue manager

  Example:
  ```
  crtmqm -u SYSTEM.DEAD.LETTER.QUEUE QMGR01
  ```

Option 2:

1. Identify a user-defined dead-letter queue when you create the queue manager

   Example:
   ```
   crtmqm -u QMGR01.DLQ QMGR01
   ```

2. Create dead-letter queue as a local queue on the queue manager

   Example:
   ```
   runmqsc QMGR01
    DEF QL(QMGR01.DLQ)
    END
   ```

Option 3:

1. Create queue manager and do not identify a dead-letter queue
2. Create a local queue and then alter queue manager to use local queue for the dead-letter queue

   Example:
   ```
    DEF QL(APP.DLQ)
    ALTER QMGR DEADQ(APP.DLQ)
   ```

Creating a queue manager and queues © Copyright IBM Corporation 2017

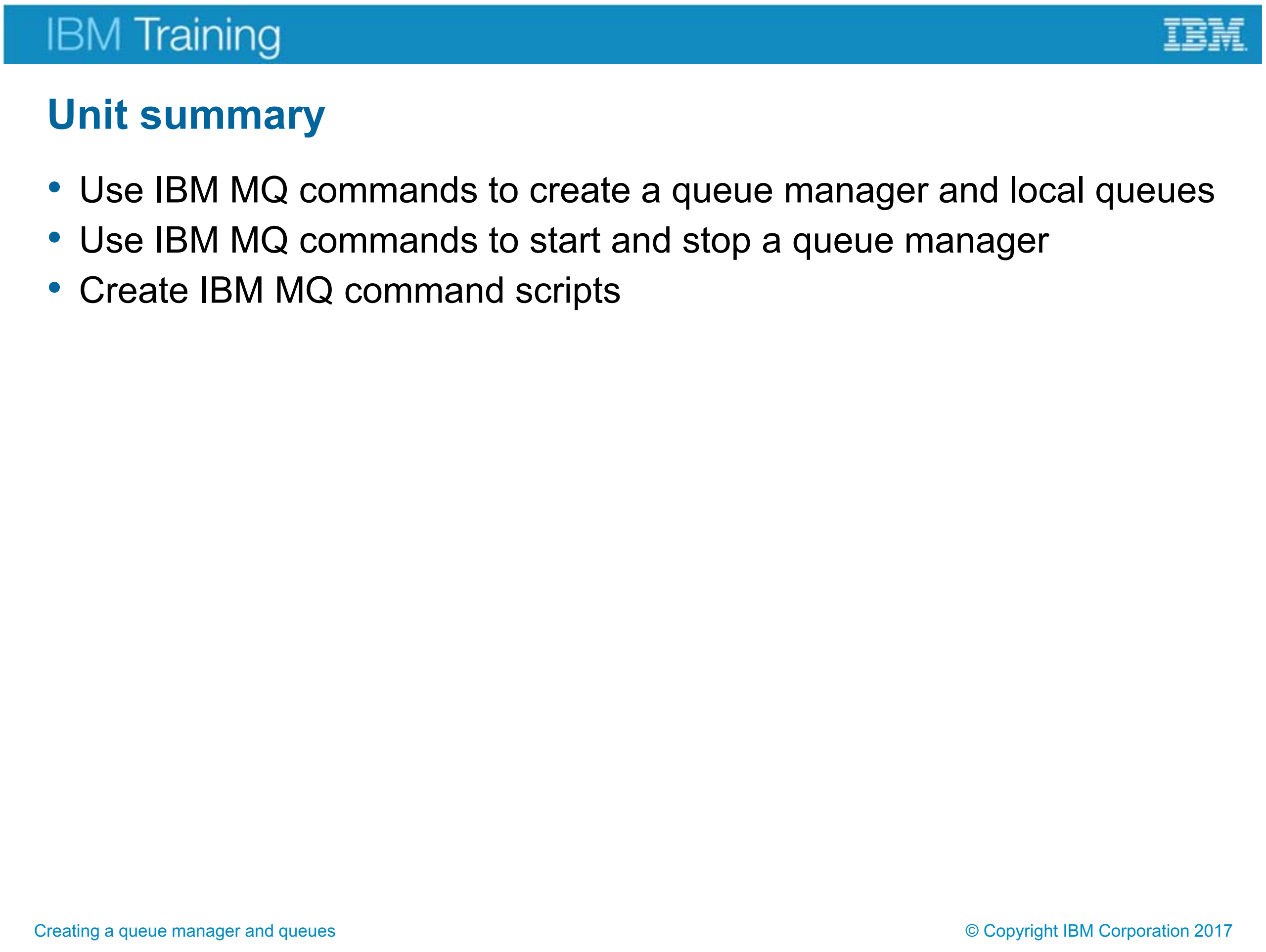
Figure 3-32. Defining a dead-letter queue

You can define a dead-letter queue for a queue manager by using one of three methods.

The first option is to identify the dead-letter queue when you create the queue manager with the **-u** option. In the example, the command that creates the queue manager that is named **QMGR01** identifies the SYSTEM dead-letter queue that is named **SYSTEM.DEAD.LETTER.QUEUE** as its dead-letter queue.

The second option is to identify a user-defined dead-letter queue when you create the queue manager with the **-u** option. In the example, the command that creates the queue manager that is named **QMGR01** identifies the dead-letter queue that is named **QMGR01.DLQ** as its dead-letter queue. Then, after the queue manager is created, the user-defined queue is created on the queue manager by using the **DEF QL** command.

The third option is to assign the dead-letter queue after the queue manager is created by using the MQSC **ALTER QMGR** command. In this third option, step 2 creates a local queue that is named **APP.DLQ** and then modifies the queue manager **DEADQ** attribute.



IBM Training

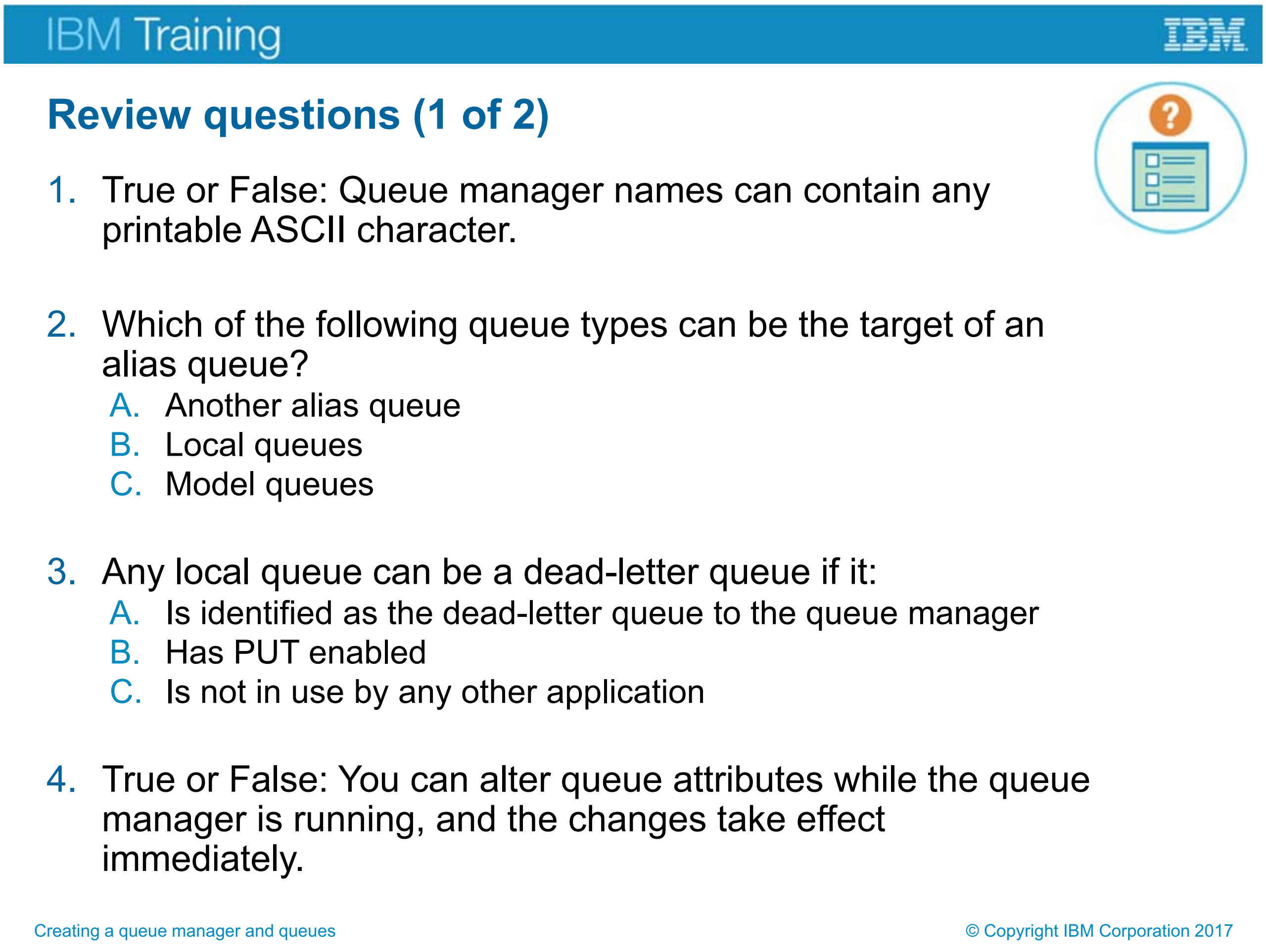
Unit summary

- Use IBM MQ commands to create a queue manager and local queues
- Use IBM MQ commands to start and stop a queue manager
- Create IBM MQ command scripts

Creating a queue manager and queues

© Copyright IBM Corporation 2017

Figure 3-33. Unit summary



IBM Training

Review questions (1 of 2)

1. True or False: Queue manager names can contain any printable ASCII character.
2. Which of the following queue types can be the target of an alias queue?
   A. Another alias queue
   B. Local queues
   C. Model queues
3. Any local queue can be a dead-letter queue if it:
   A. Is identified as the dead-letter queue to the queue manager
   B. Has PUT enabled
   C. Is not in use by any other application
4. True or False: You can alter queue attributes while the queue manager is running, and the changes take effect immediately.

Creating a queue manager and queues © Copyright IBM Corporation 2017

Figure 3-34. Review questions (1 of 2)

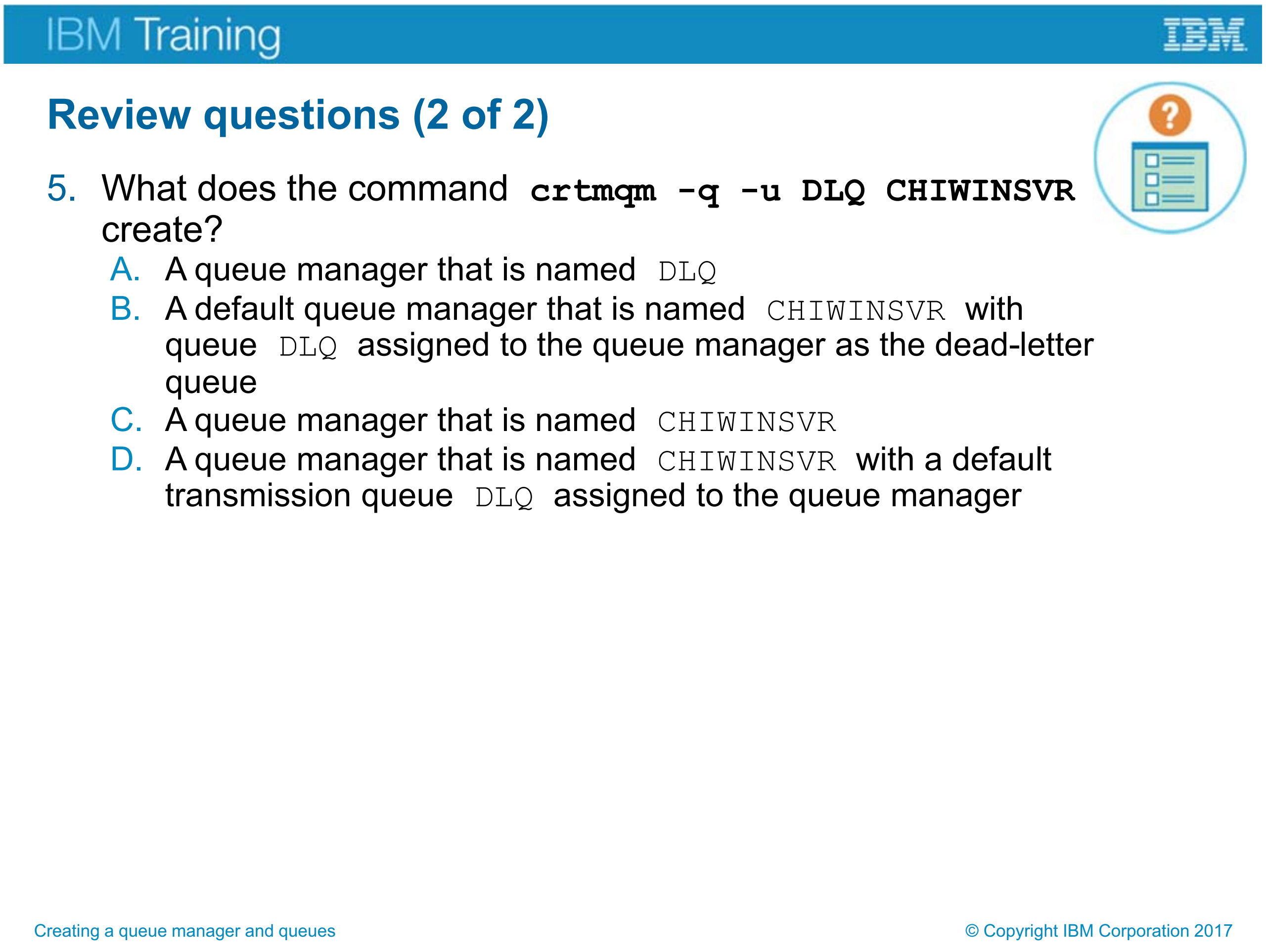
Write your answers here:

1.

2.

3.

4.



IBM Training

Review questions (2 of 2)

5. What does the command **`crtmqm -q -u DLQ CHIWINSVR`** create?
   A. A queue manager that is named `DLQ`
   B. A default queue manager that is named `CHIWINSVR` with queue `DLQ` assigned to the queue manager as the dead-letter queue
   C. A queue manager that is named `CHIWINSVR`
   D. A queue manager that is named `CHIWINSVR` with a default transmission queue `DLQ` assigned to the queue manager

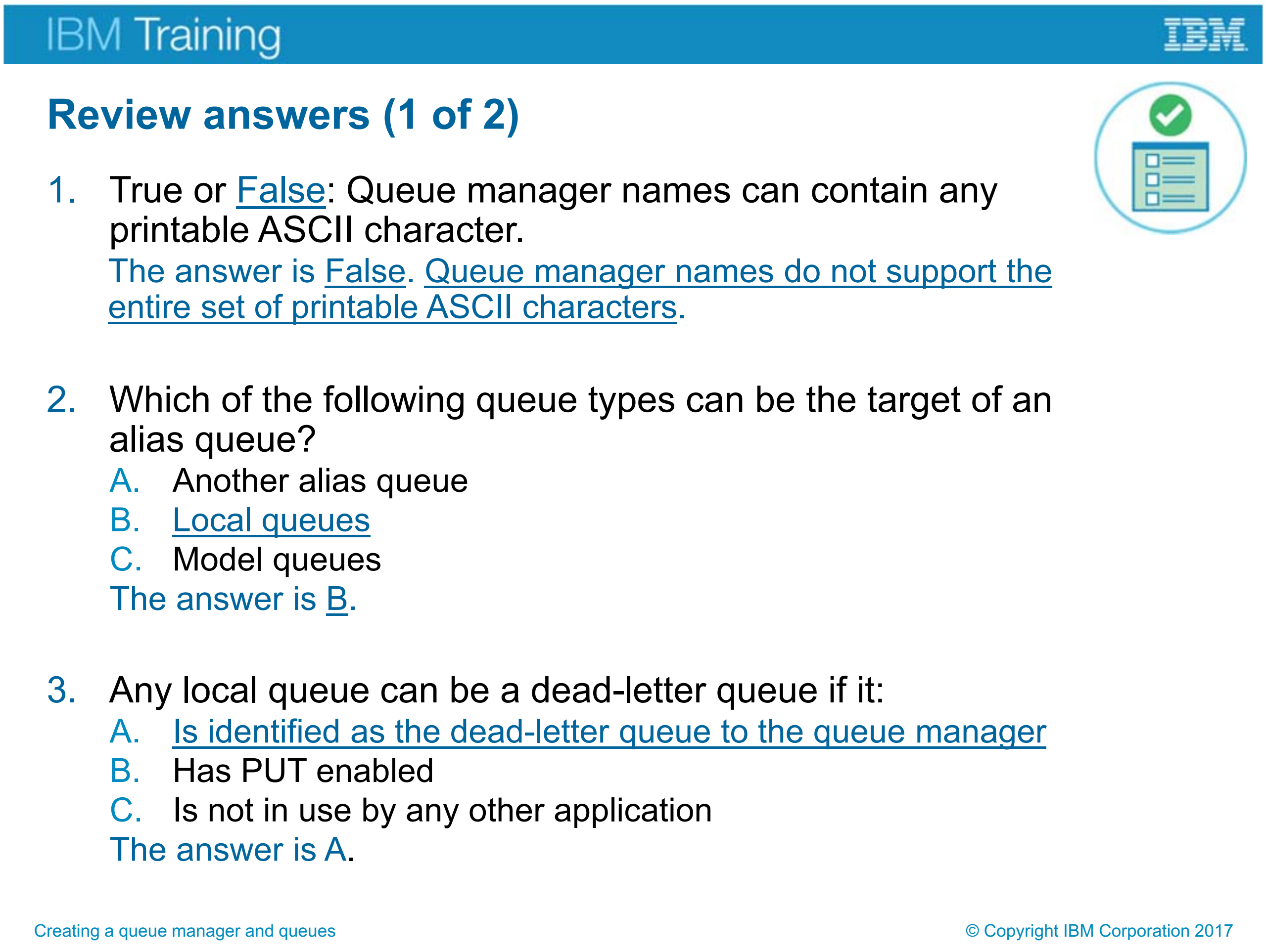
Creating a queue manager and queues

© Copyright IBM Corporation 2017

Figure 3-35. Review questions (2 of 2)

Write your answer here:

5.



IBM Training

Review answers (1 of 2)

1. True or False: Queue manager names can contain any printable ASCII character.
   The answer is False. Queue manager names do not support the entire set of printable ASCII characters.

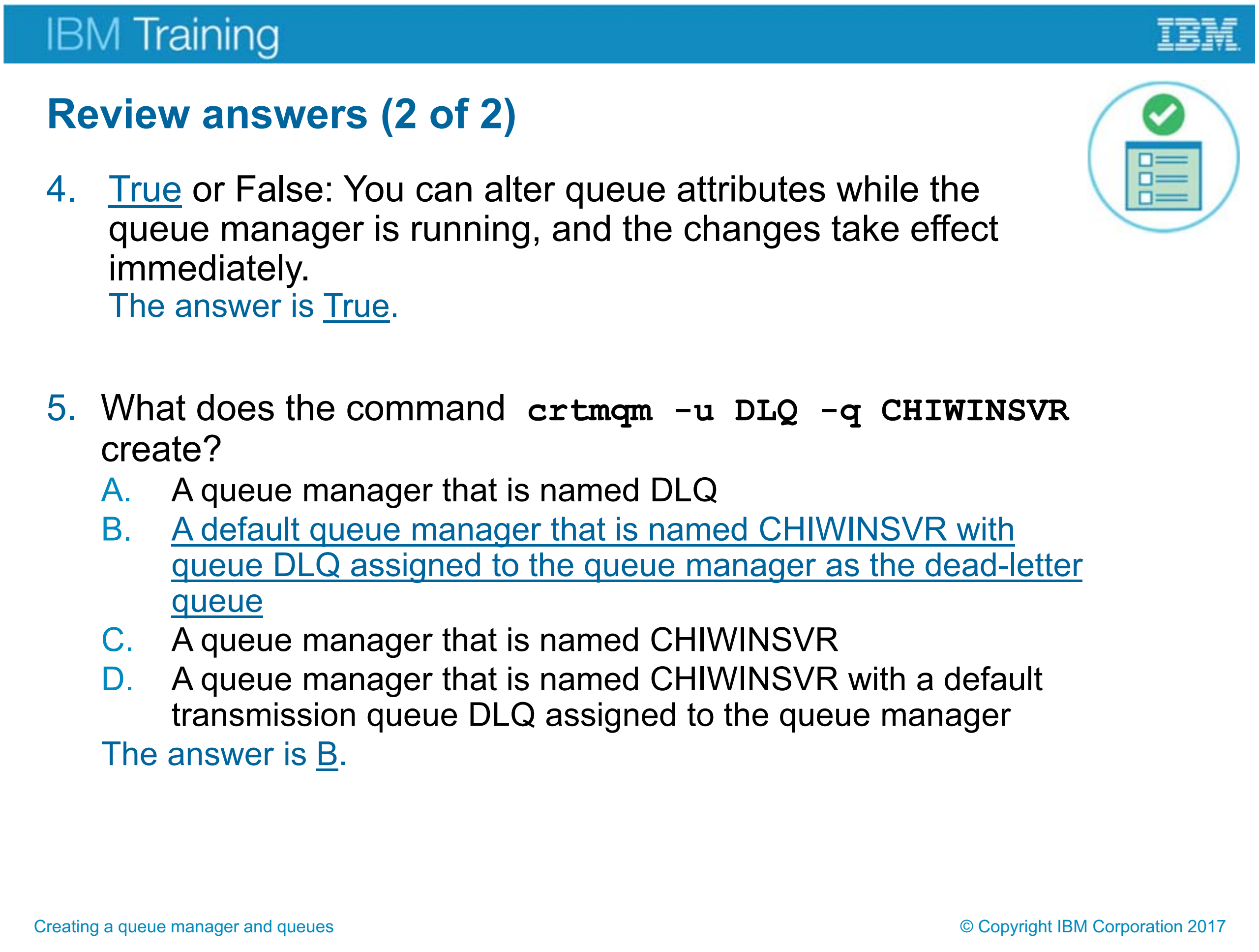
2. Which of the following queue types can be the target of an alias queue?
   A. Another alias queue
   B. Local queues
   C. Model queues
   The answer is B.

3. Any local queue can be a dead-letter queue if it:
   A. Is identified as the dead-letter queue to the queue manager
   B. Has PUT enabled
   C. Is not in use by any other application
   The answer is A.

Creating a queue manager and queues

© Copyright IBM Corporation 2017

Figure 3-36. Review answers (1 of 2)



IBM Training

Review answers (2 of 2)

4. True or False: You can alter queue attributes while the queue manager is running, and the changes take effect immediately.
   The answer is True.

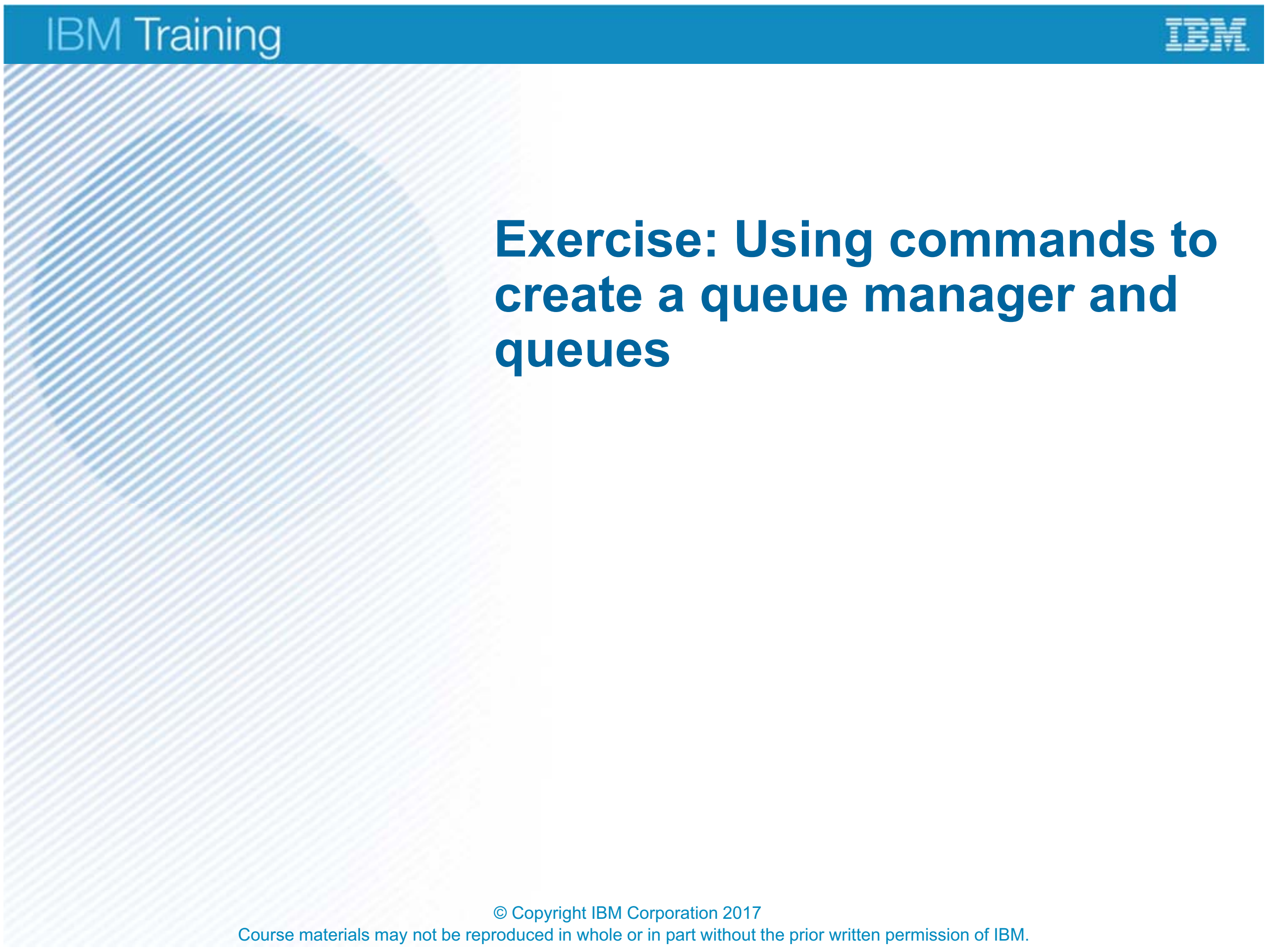
5. What does the command `crtmqm -u DLQ -q CHIWINSVR` create?
   A. A queue manager that is named DLQ
   B. A default queue manager that is named CHIWINSVR with queue DLQ assigned to the queue manager as the dead-letter queue
   C. A queue manager that is named CHIWINSVR
   D. A queue manager that is named CHIWINSVR with a default transmission queue DLQ assigned to the queue manager

   The answer is B.

Creating a queue manager and queues

© Copyright IBM Corporation 2017

Figure 3-37. Review answers (2 of 2)

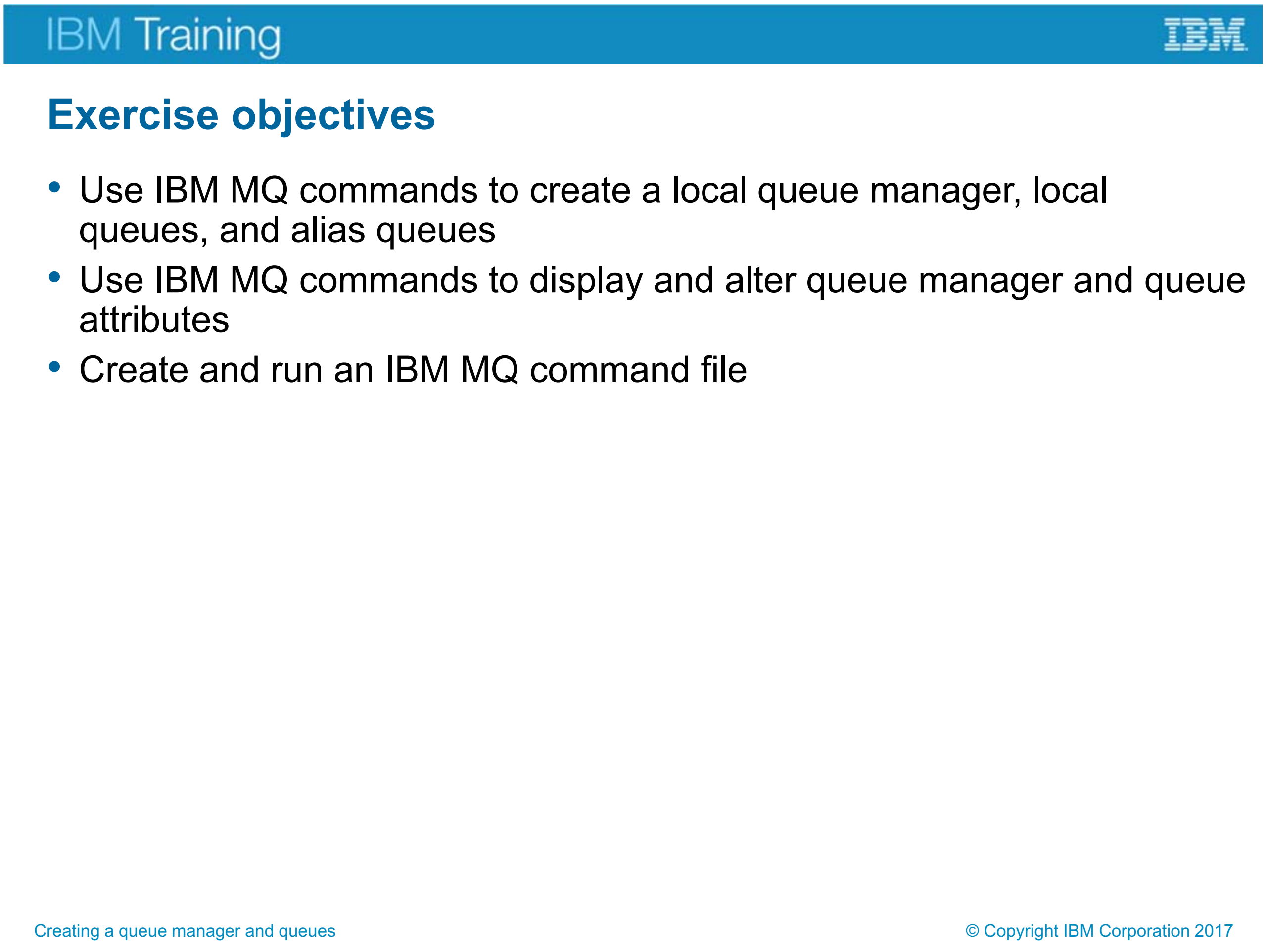


IBM Training

Exercise: Using commands to create a queue manager and queues

Figure 3-38. Exercise: Using commands to create queue managers and queues

In this exercise, you create a queue manager, start it, and then create queues.



IBM Training

Exercise objectives

- Use IBM MQ commands to create a local queue manager, local queues, and alias queues
- Use IBM MQ commands to display and alter queue manager and queue attributes
- Create and run an IBM MQ command file

Creating a queue manager and queues

© Copyright IBM Corporation 2017

Figure 3-39. Exercise objectives

See the *Course Exercises Guide* for the exercise description and detailed instructions.

Unit 4. Introduction to IBM MQ Explorer

Estimated time

00:30

Overview

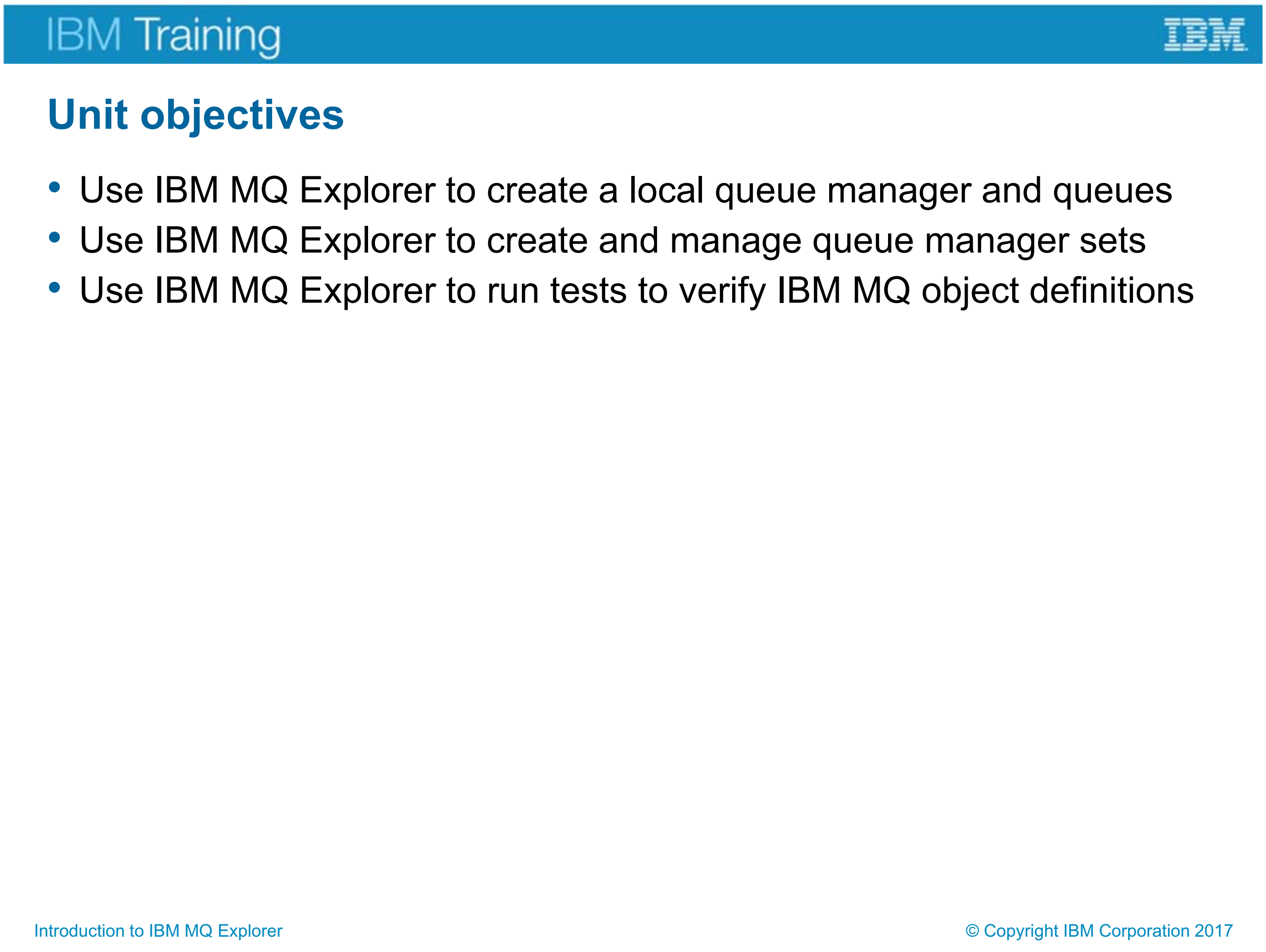
IBM MQ Explorer is a graphical interface for administering IBM MQ. In this unit, you learn how to use IBM MQ Explorer to administer IBM MQ objects.

How you will check your progress

- Review questions
- Hands-on exercise

References

IBM Knowledge Center for IBM MQ v9



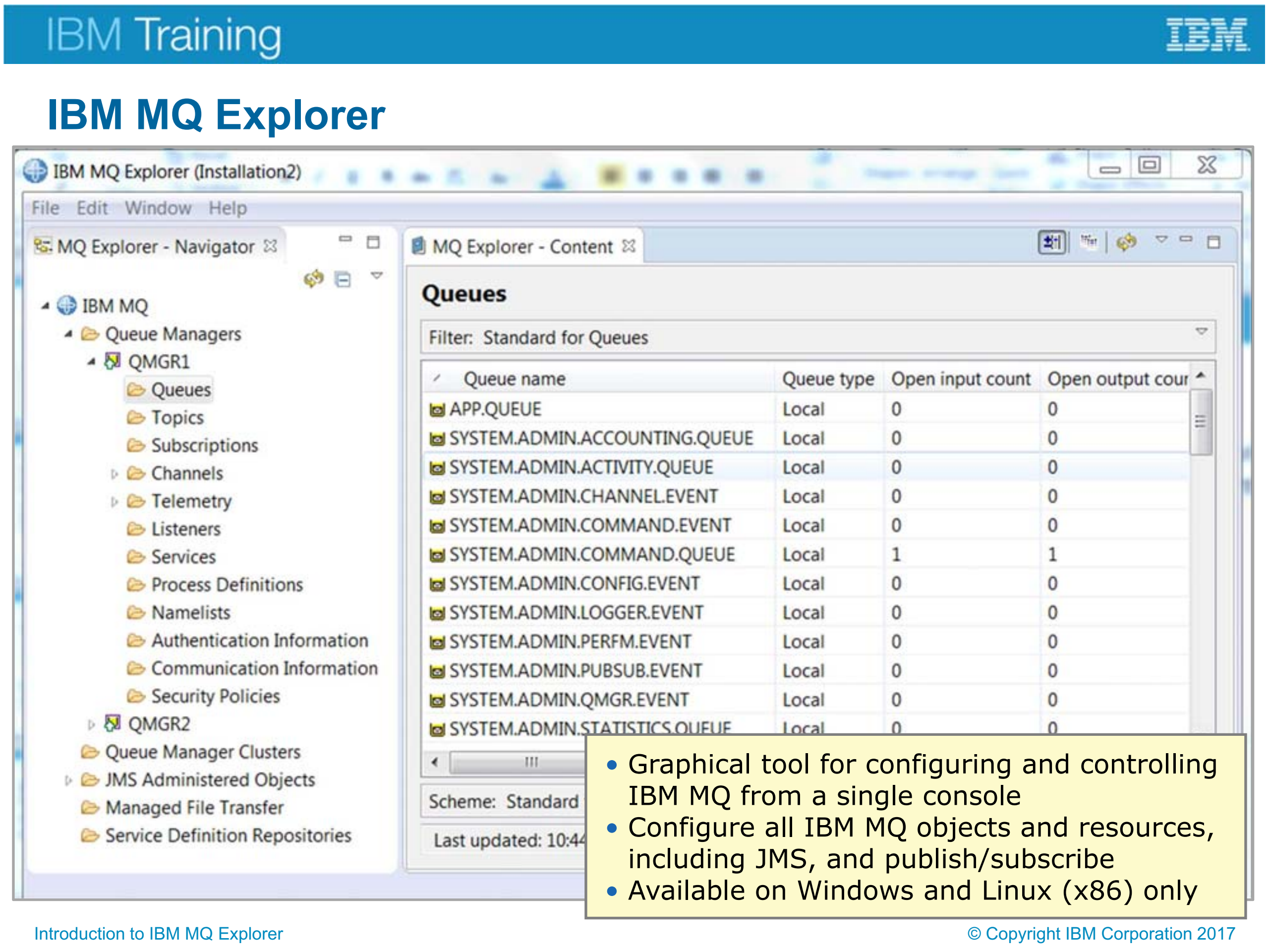
IBM Training

Unit objectives

- Use IBM MQ Explorer to create a local queue manager and queues
- Use IBM MQ Explorer to create and manage queue manager sets
- Use IBM MQ Explorer to run tests to verify IBM MQ object definitions

Introduction to IBM MQ Explorer

Figure 4-1. Unit objectives



IBM MQ Explorer

Queue name	Queue type	Open input count	Open output cour
APP.QUEUE	Local	0	0
SYSTEM.ADMIN.ACCOUNTING.QUEUE	Local	0	0
SYSTEM.ADMIN.ACTIVITY.QUEUE	Local	0	0
SYSTEM.ADMIN.CHANNEL.EVENT	Local	0	0
SYSTEM.ADMIN.COMMAND.EVENT	Local	0	0
SYSTEM.ADMIN.COMMAND.QUEUE	Local	1	1
SYSTEM.ADMIN.CONFIG.EVENT	Local	0	0
SYSTEM.ADMIN.LOGGER.EVENT	Local	0	0
SYSTEM.ADMIN.PERFM.EVENT	Local	0	0
SYSTEM.ADMIN.PUBSUB.EVENT	Local	0	0
SYSTEM.ADMIN.QMGR.EVENT	Local	0	0
SYSTEM.ADMIN.STATISTICS.QUEUE	Local	0	0

- Graphical tool for configuring and controlling IBM MQ from a single console
- Configure all IBM MQ objects and resources, including JMS, and publish/subscribe
- Available on Windows and Linux (x86) only

Introduction to IBM MQ Explorer © Copyright IBM Corporation 2017

Figure 4-2. IBM MQ Explorer

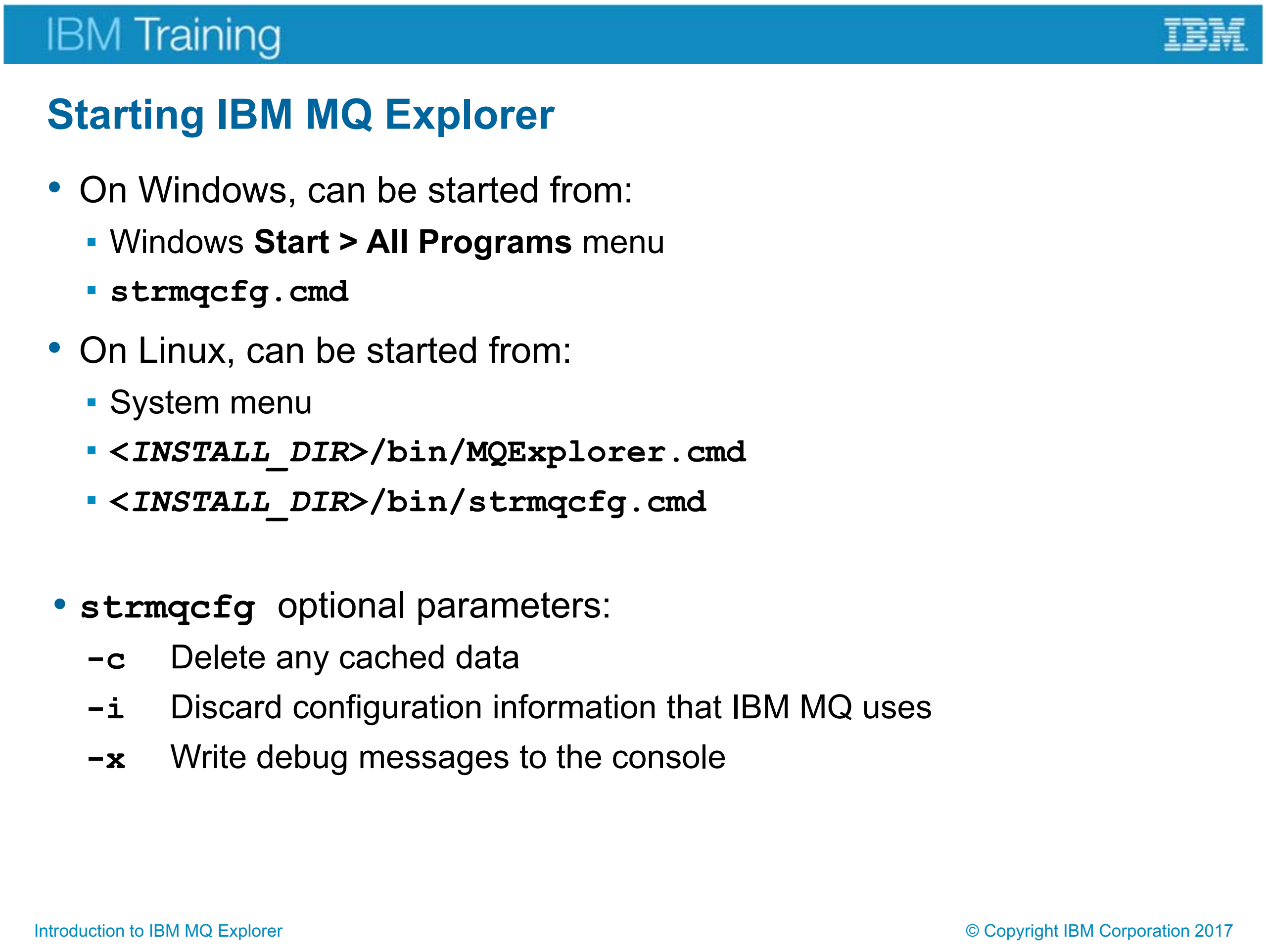
IBM MQ Explorer is the graphical user interface for administering and monitoring IBM MQ components, whether your local computer or a remote server hosts them.

MQ Explorer also supports:

- Advanced filtering
- Management of application connections
- Grouping of queue managers to simplify management
- Publish/subscribe administration

MQ Explorer is built as a rich client platform (RCP) application on Eclipse. It runs on Windows and Linux (x86) operating systems but can be used to remotely manage MQ objects on other operating systems.

All functions that were provided in the previous release are available with the new version of MQ.



IBM Training

Starting IBM MQ Explorer

- On Windows, can be started from:
  - Windows **Start > All Programs** menu
  - **`strmqcfg.cmd`**
- On Linux, can be started from:
  - System menu
  - **`<INSTALL_DIR>/bin/MQExplorer.cmd`**
  - **`<INSTALL_DIR>/bin/strmqcfg.cmd`**

- **`strmqcfg`** optional parameters:

`-c`	Delete any cached data
`-i`	Discard configuration information that IBM MQ uses
`-x`	Write debug messages to the console

Introduction to IBM MQ Explorer © Copyright IBM Corporation 2017

Figure 4-3. Starting IBM MQ Explorer

IBM MQ Explorer can be started from a command or from the system **Start** menu.

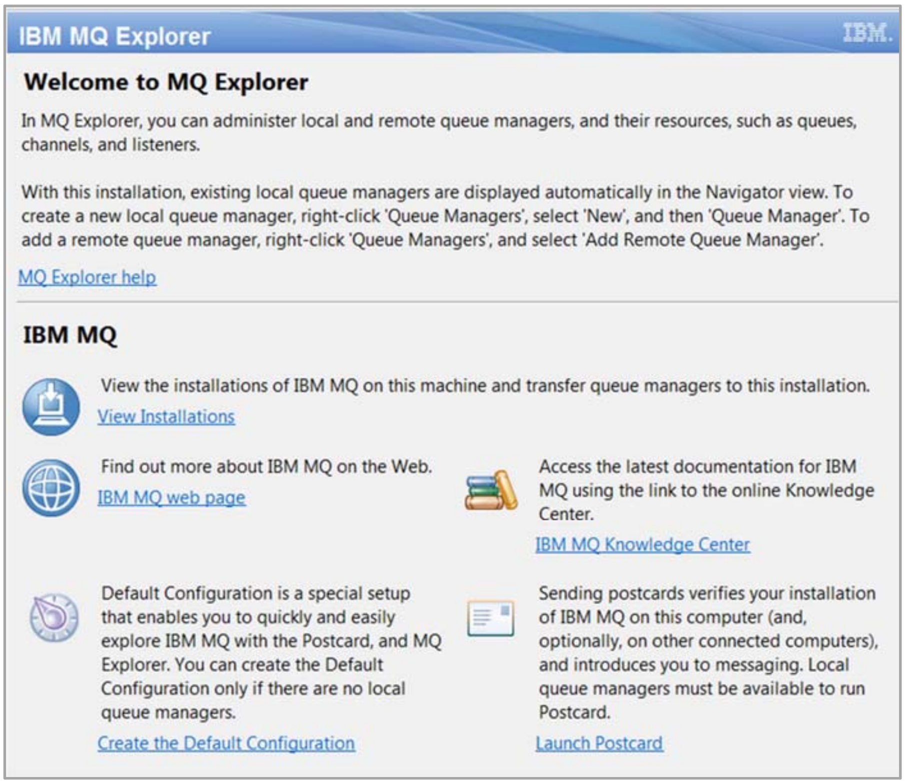
On both Windows and Linux, the command to start MQ Explorer is **`strmqcfg`**.

The **`strmqcfg`** command has three optional parameters to clear the cache and generate debug messages to the console.

On Linux, MQ Explorer can be started from the **System** menu that is based on Linux provider.

IBM Training

Welcome page



IBM MQ Explorer

Welcome to MQ Explorer

In MQ Explorer, you can administer local and remote queue managers, and their resources, such as queues, channels, and listeners.

With this installation, existing local queue managers are displayed automatically in the Navigator view. To create a new local queue manager, right-click 'Queue Managers', select 'New', and then 'Queue Manager'. To add a remote queue manager, right-click 'Queue Managers', and select 'Add Remote Queue Manager'.

MQ Explorer help

IBM MQ

View the installations of IBM MQ on this machine and transfer queue managers to this installation.
View Installations

Find out more about IBM MQ on the Web.
IBM MQ web page

Access the latest documentation for IBM MQ using the link to the online Knowledge Center.
IBM MQ Knowledge Center

Default Configuration is a special setup that enables you to quickly and easily explore IBM MQ with the Postcard, and MQ Explorer. You can create the Default Configuration only if there are no local queue managers.
Create the Default Configuration

Sending postcards verifies your installation of IBM MQ on this computer (and, optionally, on other connected computers), and introduces you to messaging. Local queue managers must be available to run Postcard.
Launch Postcard

- View installations of IBM MQ on this computer and optionally transfer a queue manager to a different installation
- Connect to the IBM MQ web page
- Start the IBM Knowledge Center for IBM MQ
- Create the default configuration
- Verify the installation by using the Postcard application

Introduction to IBM MQ Explorer © Copyright IBM Corporation 2017

Figure 4-4. Welcome page

The IBM MQ Explorer **Welcome** page is displayed the first time that MQ Explorer starts.

As shown in the figure, the **Welcome** page contains links for information about IBM MQ, for creating the default configuration, and verifying the installation.

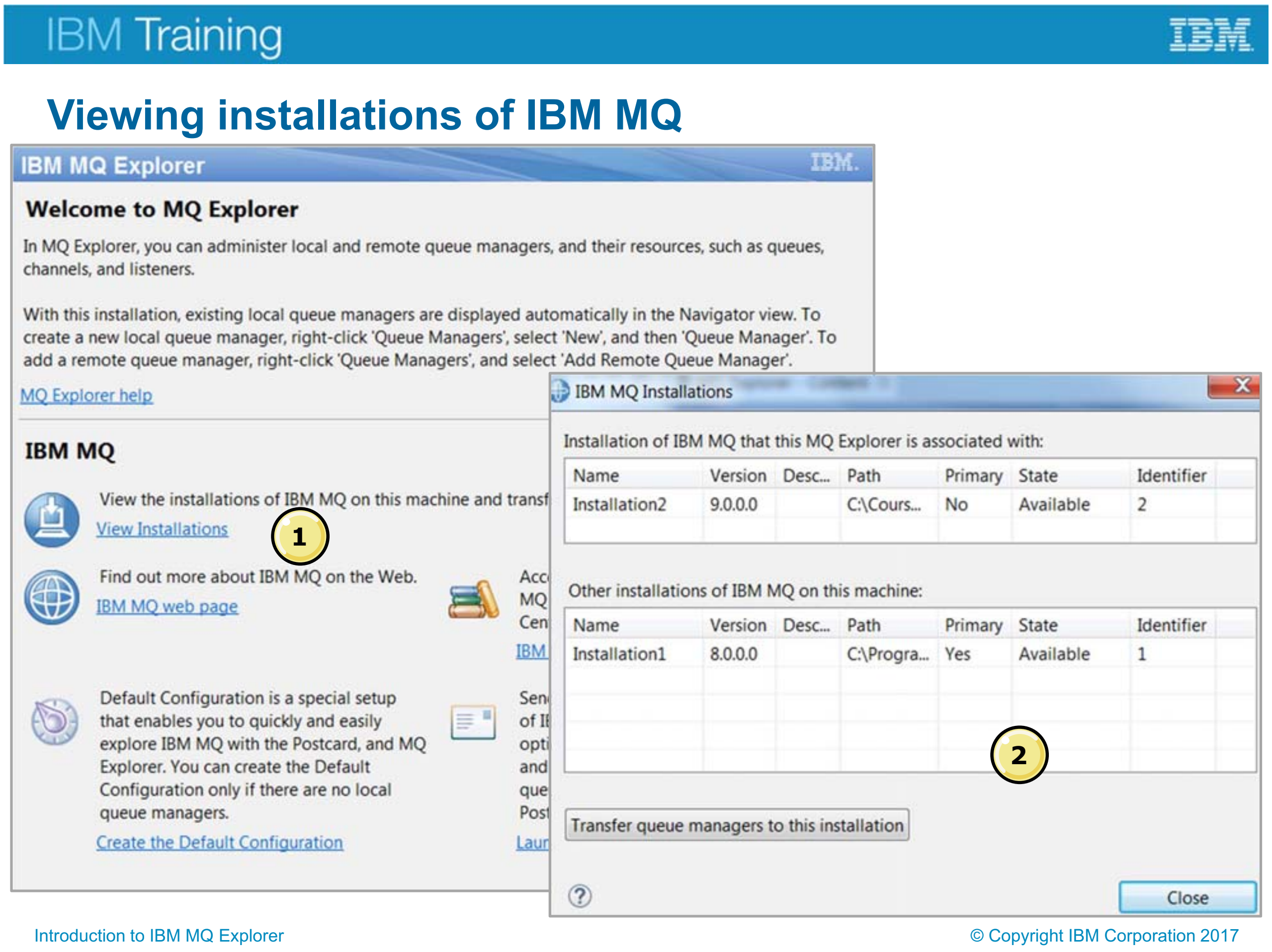
The **Welcome** page can be accessed at any time by clicking **IBM MQ** in the **MQ Explorer - Navigator** view.

You can use the **Default Configuration** wizard to add the first configured queue manager to this system. You can use the Default Configuration wizard to create, view, or alter the default configuration. You can also use this wizard to alter or display details of an existing queue manager that was created in the default configuration.

If you migrated existing queue managers, or created any queue managers since installing MQ, you cannot run the **Default Configuration** wizard.

Start the **Default Configuration** wizard by selecting **Create the Default Configuration** on the **Welcome** page.

For a new installation of IBM MQ, you can create a default configuration to explore features of MQ by using the Postcard application and MQ Explorer. The Postcard application provides a fast and simple way to verify that your MQ installation was successful. It uses the default queue manager that is created in the **Default Configuration** wizard.



IBM Training

Viewing installations of IBM MQ

IBM MQ Explorer

Welcome to MQ Explorer

In MQ Explorer, you can administer local and remote queue managers, and their resources, such as queues, channels, and listeners.

With this installation, existing local queue managers are displayed automatically in the Navigator view. To create a new local queue manager, right-click 'Queue Managers', select 'New', and then 'Queue Manager'. To add a remote queue manager, right-click 'Queue Managers', and select 'Add Remote Queue Manager'.

MQ Explorer help

IBM MQ

View the installations of IBM MQ on this machine and transf

View Installations

Find out more about IBM MQ on the Web.

IBM MQ web page

Default Configuration is a special setup that enables you to quickly and easily explore IBM MQ with the Postcard, and MQ Explorer. You can create the Default Configuration only if there are no local queue managers.

Create the Default Configuration

IBM MQ Installations

Installation of IBM MQ that this MQ Explorer is associated with:

Name	Version	Desc...	Path	Primary	State	Identifier
Installation2	9.0.0.0		C:\Cours...	No	Available	2

Other installations of IBM MQ on this machine:

Name	Version	Desc...	Path	Primary	State	Identifier
Installation1	8.0.0.0		C:\Progra...	Yes	Available	1

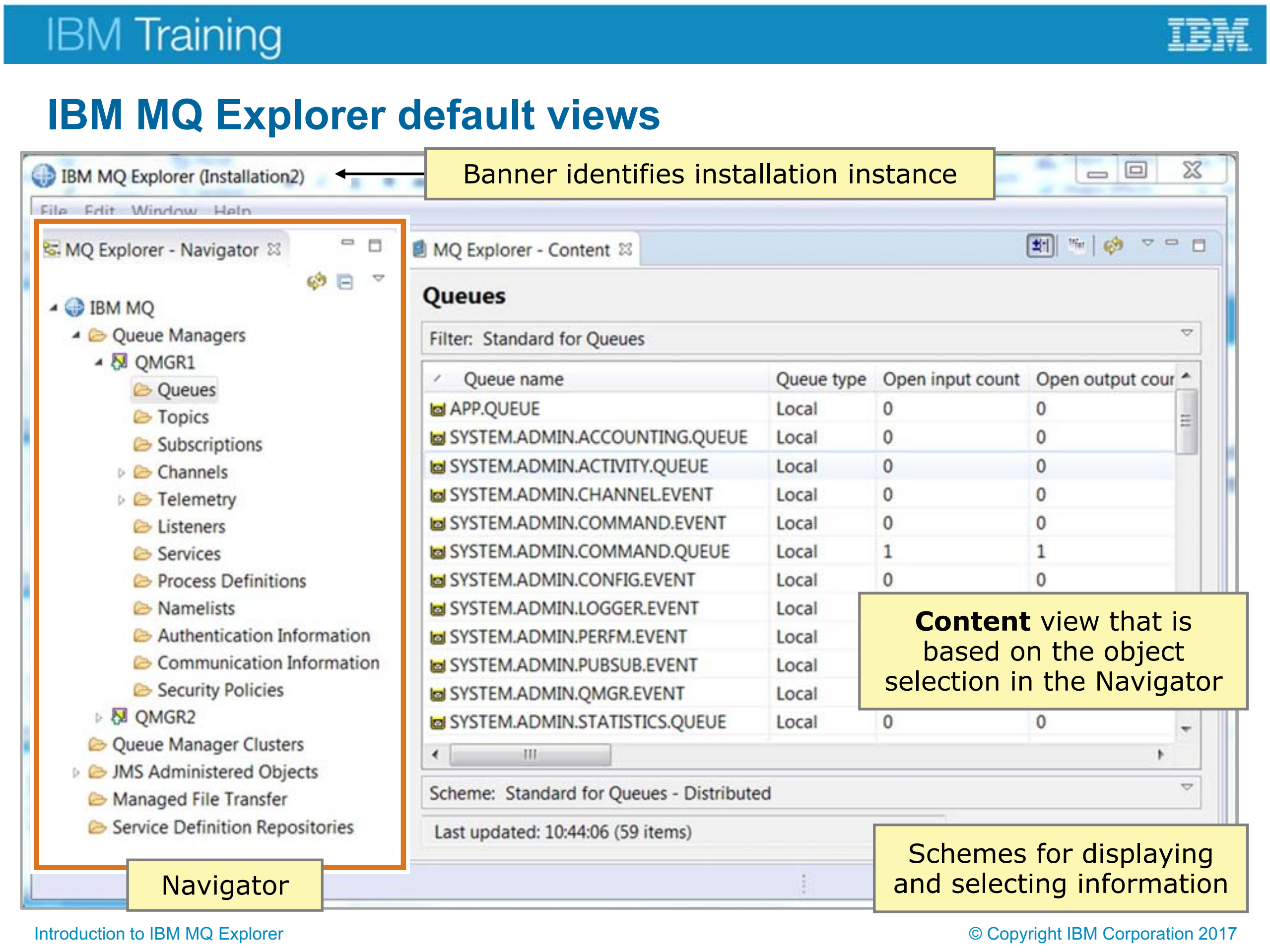
Introduction to IBM MQ Explorer © Copyright IBM Corporation 2017

Figure 4-5. Viewing installations of IBM MQ

The Welcome page in MQ Explorer contains a link for viewing the MQ installations and optionally, transferring a queue manager to a different installation.

To view the MQ installations:

1. Click **View Installations** on the **Welcome** page.
2. The IBM MQ Installations page identifies the installation with which this MQ Explorer is associated. It also lists other installations of MQ on the same computer.



IBM Training

IBM MQ Explorer default views

Queues

Filter: Standard for Queues

Queue name	Queue type	Open input count	Open output cour
APP.QUEUE	Local	0	0
SYSTEM.ADMIN.ACCOUNTING.QUEUE	Local	0	0
SYSTEM.ADMIN.ACTIVITY.QUEUE	Local	0	0
SYSTEM.ADMIN.CHANNEL.EVENT	Local	0	0
SYSTEM.ADMIN.COMMAND.EVENT	Local	0	0
SYSTEM.ADMIN.COMMAND.QUEUE	Local	1	1
SYSTEM.ADMIN.CONFIG.EVENT	Local	0	0
SYSTEM.ADMIN.LOGGER.EVENT	Local		
SYSTEM.ADMIN.PERFM.EVENT	Local		
SYSTEM.ADMIN.PUBSUB.EVENT	Local		
SYSTEM.ADMIN.QMGR.EVENT	Local		
SYSTEM.ADMIN.STATISTICS.QUEUE	Local	0	0

Scheme: Standard for Queues - Distributed

Last updated: 10:44:06 (59 items)

Introduction to IBM MQ Explorer © Copyright IBM Corporation 2017

Figure 4-6. IBM MQ Explorer default views

The figure shows the general layout of the MQ Explorer user interface.

The **MQ Explorer - Navigator** view contains a list of all the objects that MQ Explorer manages. Status icons next to some of the objects, such as queue managers, indicate the status of the object. Each queue manager is listed under the **Queue Managers** folder. You can expand the queue manager to access the queue manager objects such as queues, topics, and subscriptions.

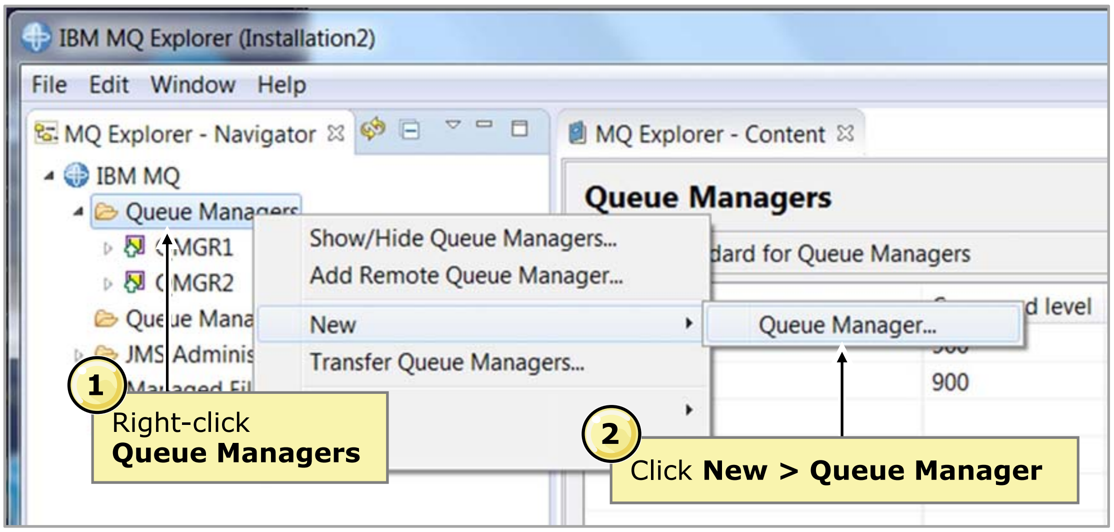
The **Content** views contain more information about the object that is selected in the **MQ Explorer - Navigator** view. For example, if the **Queues** folder under a queue manager is selected in the **MQ Explorer - Navigator** view, the **Content** view contains information about the queues on that queue manager.

You can use the **Scheme** icon (bottom portion of the **Queue** content view) to edit the current scheme and select the information to display in the property columns. You can also change the order of the columns.

The example in the figure shows the **Queues** content for the queue manager that is named QMGR1.

IBM Training

Creating a local queue manager (1 of 3)



Introduction to IBM MQ Explorer © Copyright IBM Corporation 2017

Figure 4-7. Creating a local queue manager (1 of 3)

You can create a local queue manager by using an MQ control command or by using MQ Explorer.

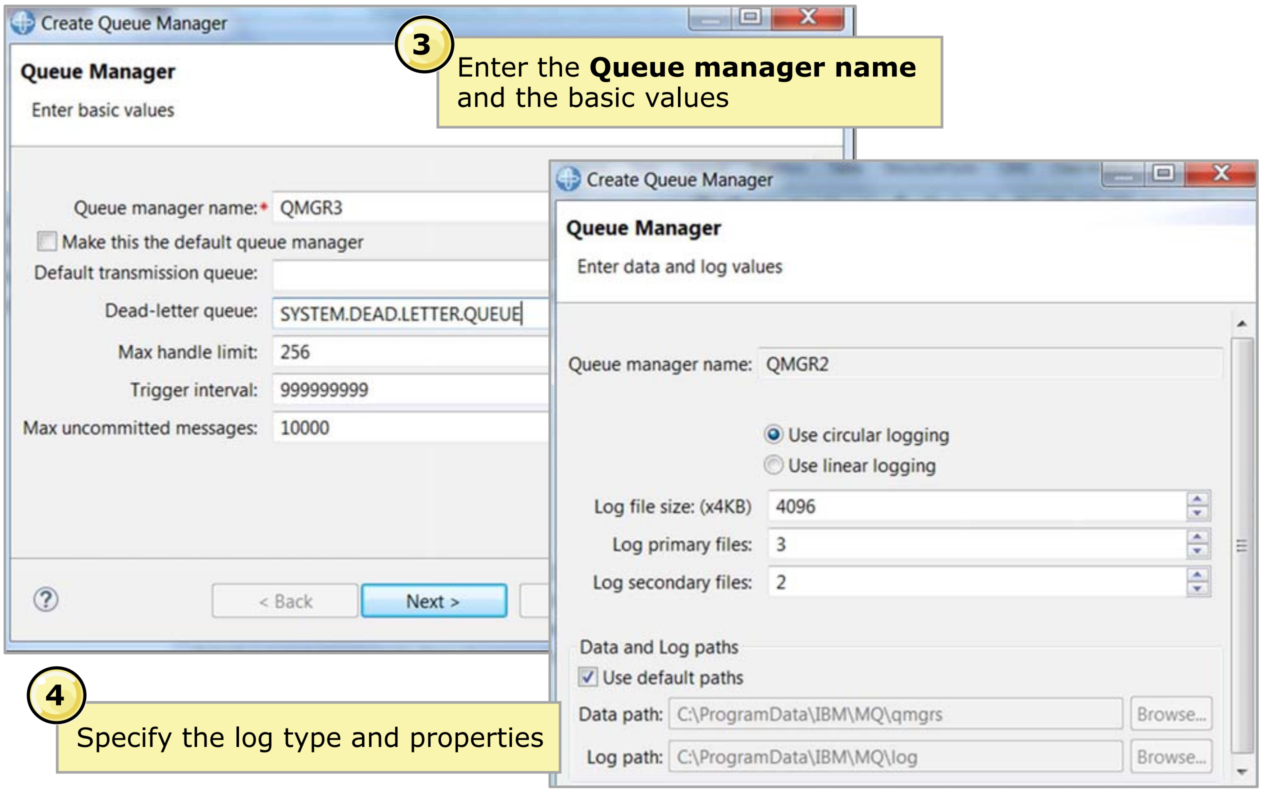
MQ Explorer automatically shows all local queue managers regardless of the method that is used to create them.

The figure shows the first two steps for creating a local queue manager by using MQ Explorer.

1. Right-click **Queue Managers** in the **MQ Explorer - Navigator** view.
2. Click **New > Queue Manager**.

IBM Training

Creating a local queue manager (2 of 3)


Create Queue Manager
3
Enter the **Queue manager name** and the basic values
4
Specify the log type and properties
< Back
Next >
Browse...
Browse...

Queue Manager

Enter basic values

Queue manager name:*	QMGR3
☐ Make this the default queue manager	
Default transmission queue:	
Dead-letter queue:	SYSTEM.DEAD.LETTER.QUEUE
Max handle limit:	256
Trigger interval:	999999999
Max uncommitted messages:	10000

Queue Manager

Enter data and log values

Queue manager name:	QMGR2
● Use circular logging	
○ Use linear logging	
Log file size: (x4KB)	4096
Log primary files:	3
Log secondary files:	2

Data and Log paths

☑ Use default paths

Data path:	C:\ProgramData\IBM\MQ\qmgrs
Log path:	C:\ProgramData\IBM\MQ\log

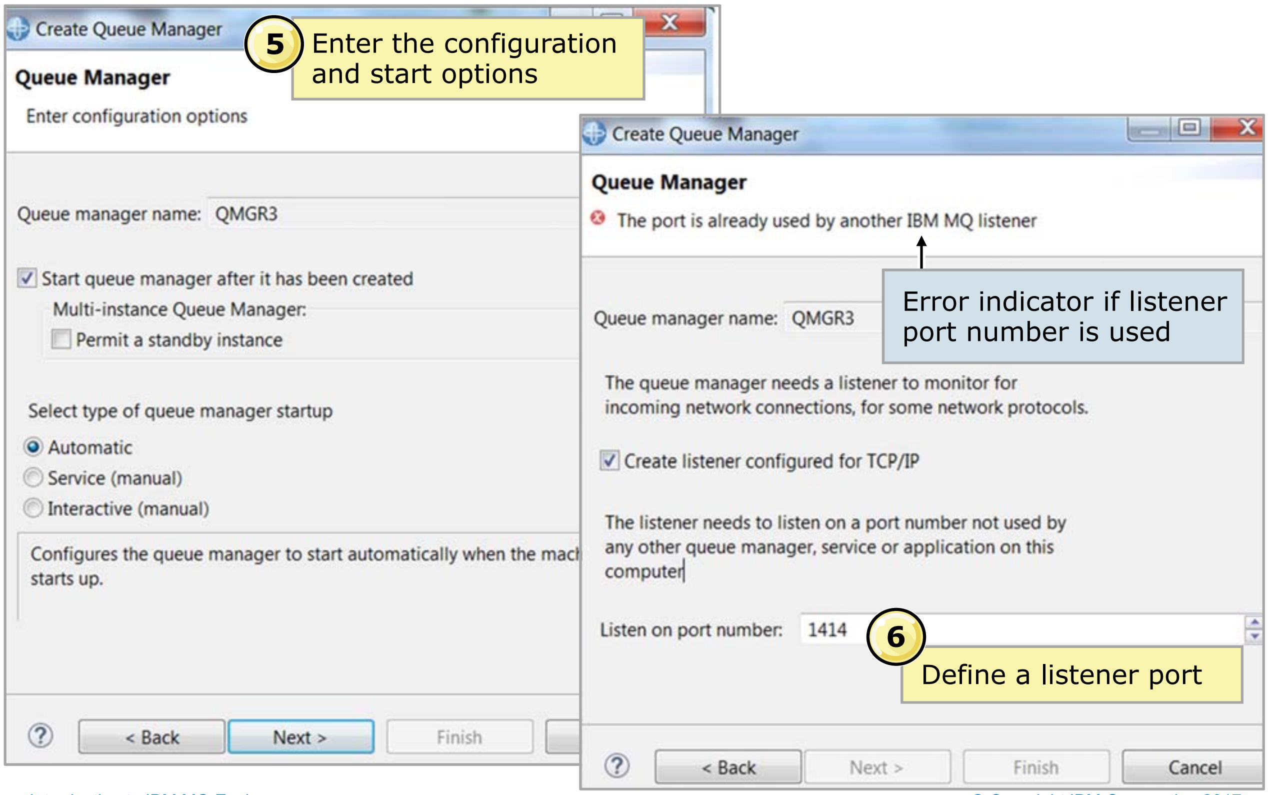
Introduction to IBM MQ Explorer © Copyright IBM Corporation 2017

Figure 4-8. Creating a local queue manager (2 of 3)

3. Specify a queue manager name and other basic configuration. Basic configuration includes specifying a default transmission queue, dead-letter queue, and trigger interval.
4. Specify the queue manager log type and properties. Queue manager log files are described in more detail throughout this course.

IBM Training

Creating a local queue manager (3 of 3)

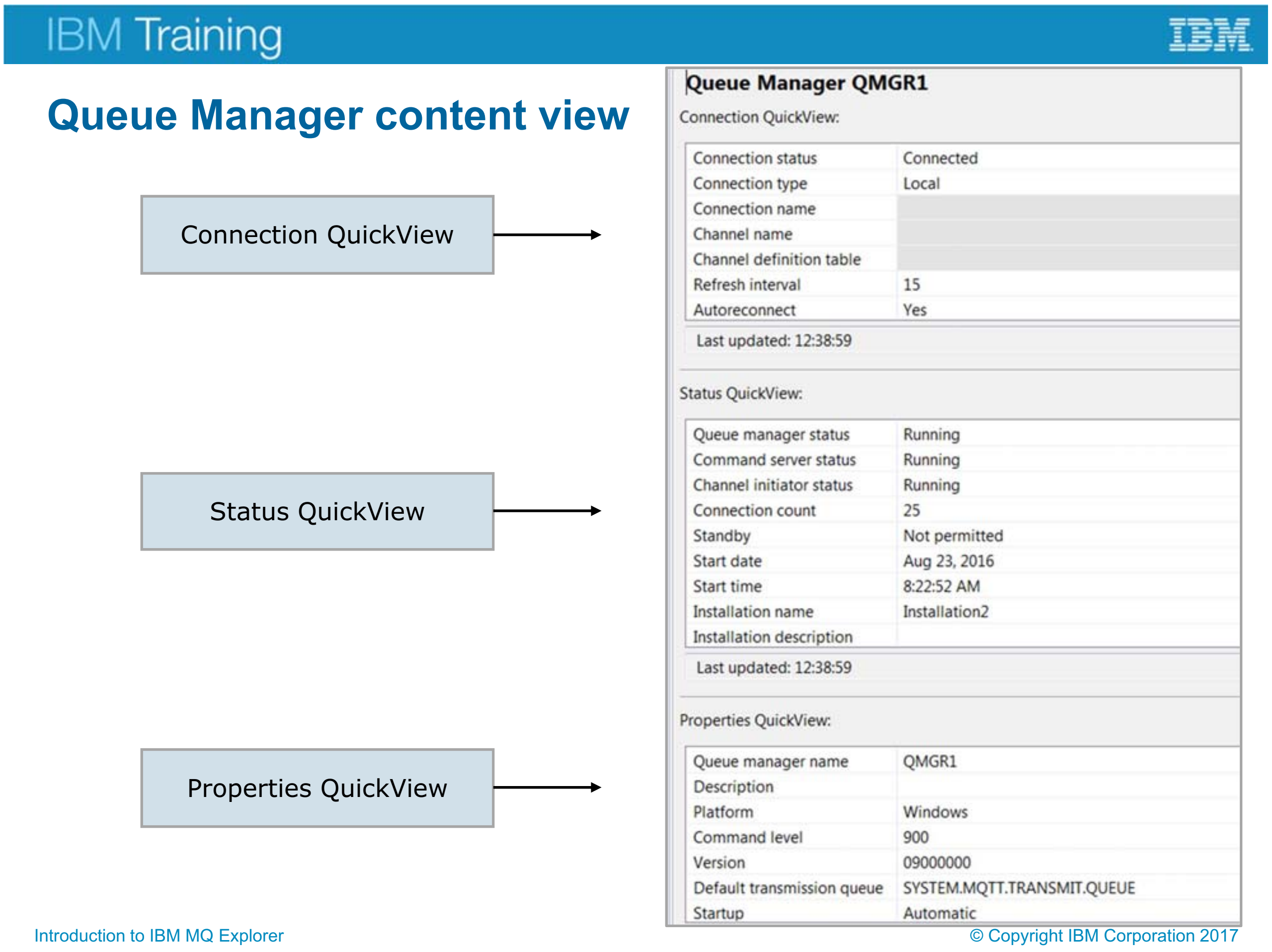

Create Queue Manager
5
Enter the configuration and start options
Queue Manager
Enter configuration options
Queue manager name: QMGR3
☑ Start queue manager after it has been created
Multi-instance Queue Manager:
☐ Permit a standby instance
Select type of queue manager startup
● Automatic
○ Service (manual)
○ Interactive (manual)
Configures the queue manager to start automatically when the mach
starts up.
< Back
Next >
Finish
Create Queue Manager
Queue Manager
The port is already used by another IBM MQ listener
Error indicator if listener port number is used
Queue manager name: QMGR3
The queue manager needs a listener to monitor for incoming network connections, for some network protocols.
☑ Create listener configured for TCP/IP
The listener needs to listen on a port number not used by any other queue manager, service or application on this computer
Listen on port number: 1414
6
Define a listener port
< Back
Next >
Finish
Cancel

Introduction to IBM MQ Explorer © Copyright IBM Corporation 2017

Figure 4-9. Creating a local queue manager (3 of 3)

5. Enter the queue manager configuration options.
   - If you want the queue manager to start immediately after it is created, select **Start queue manager after it has been created**.
   - If you want the queue manager to automatically start after a system restart, select **Automatic** for the startup type.
6. Create a listener for TCP/IP and identify the listener port value.

   A unique port number is required for each listener on the same host. If MQ detects that another listener is using the TCP/IP port number, an error message is displayed at the top of the page, as shown in the example.


IBM Training
Queue Manager content view
Connection QuickView
Status QuickView
Properties QuickView
Introduction to IBM MQ Explorer
© Copyright IBM Corporation 2017

Queue Manager QMGR1

Connection QuickView:

Connection status	Connected
Connection type	Local
Connection name	
Channel name	
Channel definition table	
Refresh interval	15
Autoreconnect	Yes

Last updated: 12:38:59

Status QuickView:

Queue manager status	Running
Command server status	Running
Channel initiator status	Running
Connection count	25
Standby	Not permitted
Start date	Aug 23, 2016
Start time	8:22:52 AM
Installation name	Installation2
Installation description	

Last updated: 12:38:59

Properties QuickView:

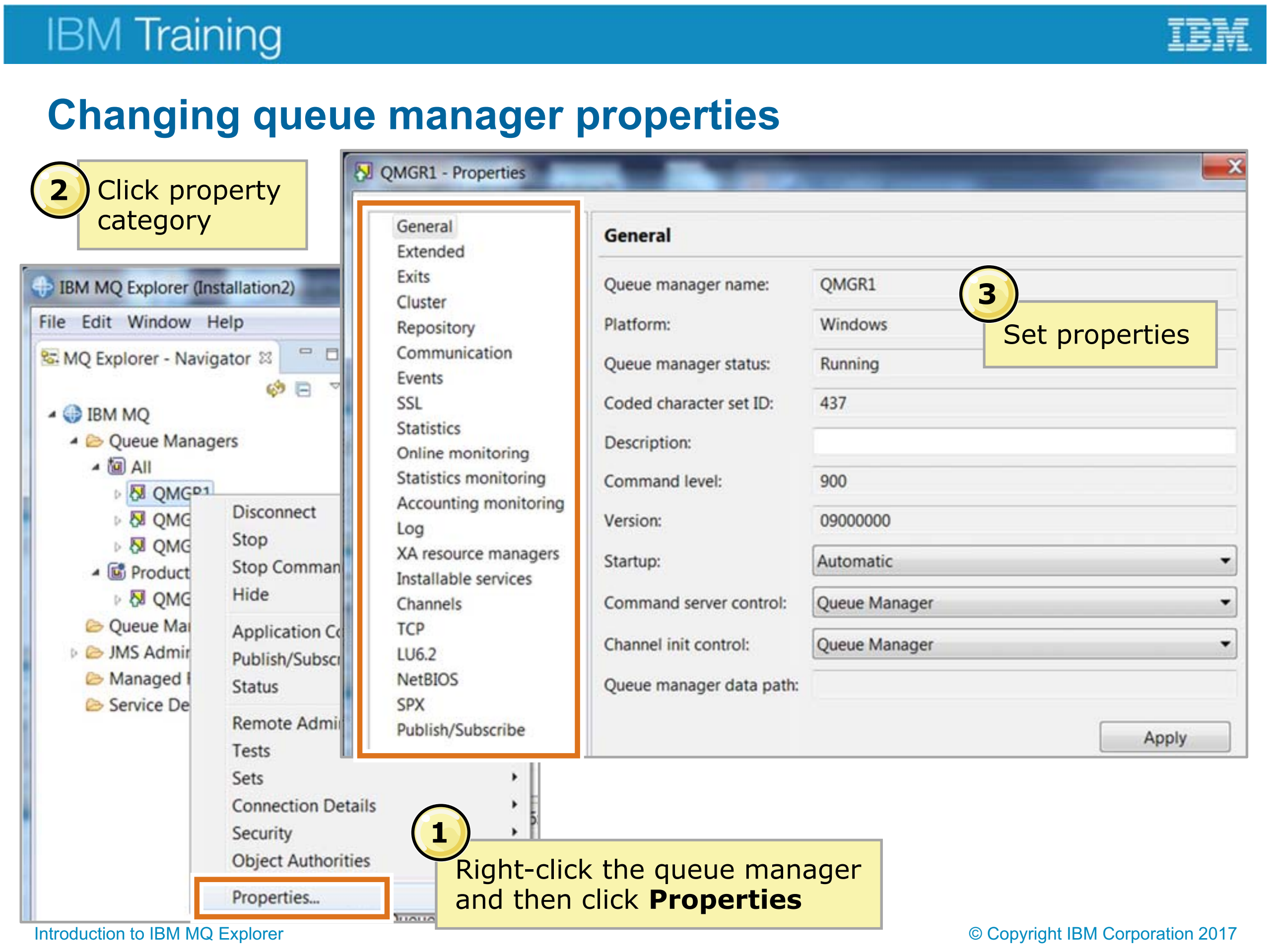
Queue manager name	QMGR1
Description	
Platform	Windows
Command level	900
Version	09000000
Default transmission queue	SYSTEM.MQTT.TRANSMIT.QUEUE
Startup	Automatic

Figure 4-10. Queue Manager content view

The **Queue Manager** content view is divided into three QuckView panes.

- **Connection QuickView** shows the queue manager connection status information, such as connection status, connection type, and connection name.
- **Status QuickView** shows the status of the queue manager. This QuickView includes the command server status, channel initiator status, and the installation name.
- **Properties QuickView** shows some of the configurable properties of the queue manager such as the operating system, command level (version of MQ), and default transmission queue.

The **Queue Manager** content view shows many of the properties that are returned with the **`DIS QMGR`** command.



Changing queue manager properties

General

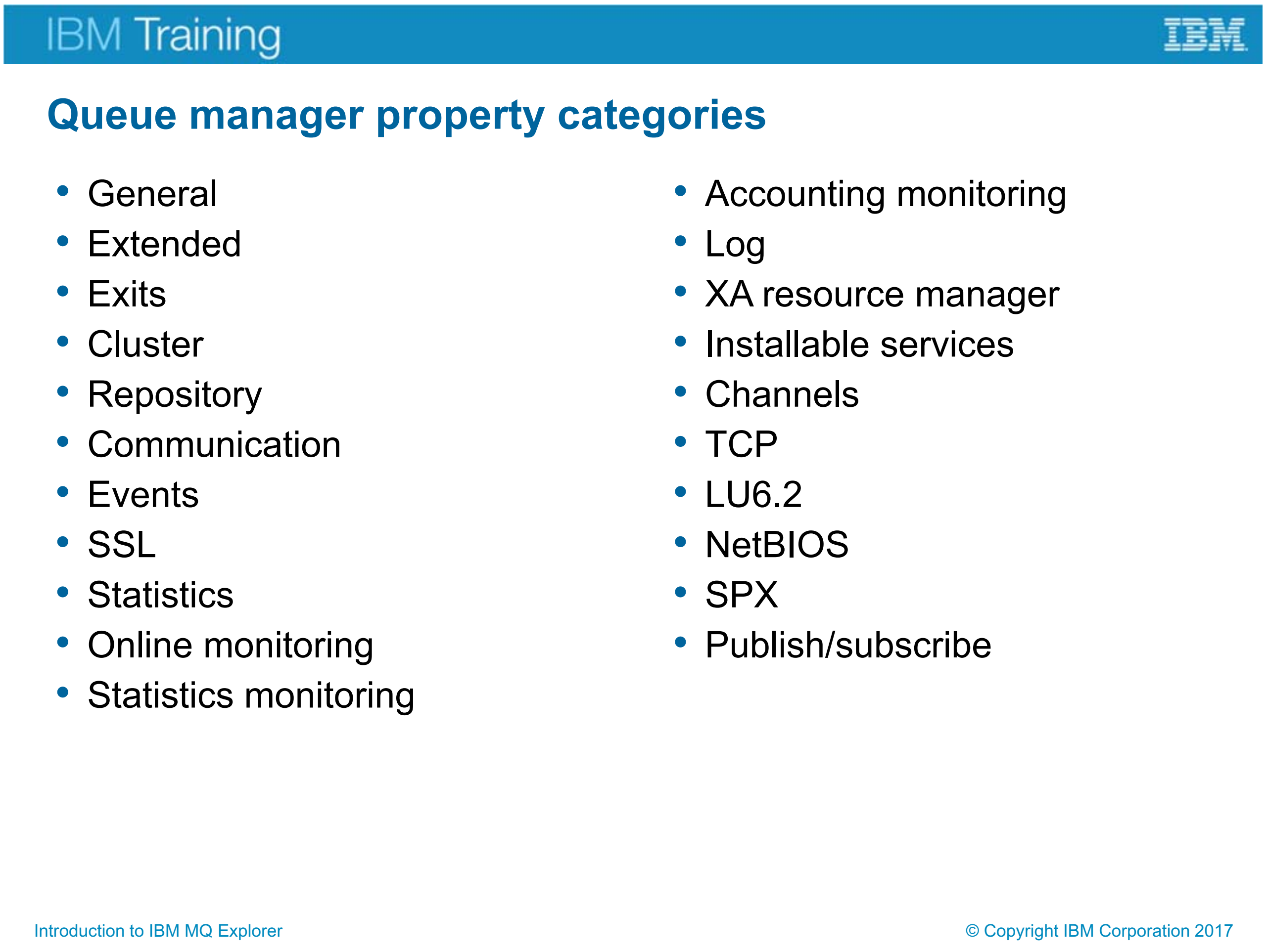
Queue manager name:	QMGR1
Platform:	Windows
Queue manager status:	Running
Coded character set ID:	437
Description:	
Command level:	900
Version:	09000000
Startup:	Automatic
Command server control:	Queue Manager
Channel init control:	Queue Manager
Queue manager data path:	

Figure 4-11. Changing queue manager properties

You can view and optionally change queue manager properties in MQ Explorer. Queue manager properties are categorized by function.

To view and change the queue manager properties:

1. Right-click the queue manager in the **MQ Explorer - Navigator** view and then click **Properties**.
2. Click the property category.
3. Change the property value and then click **Apply**.



IBM Training

Queue manager property categories

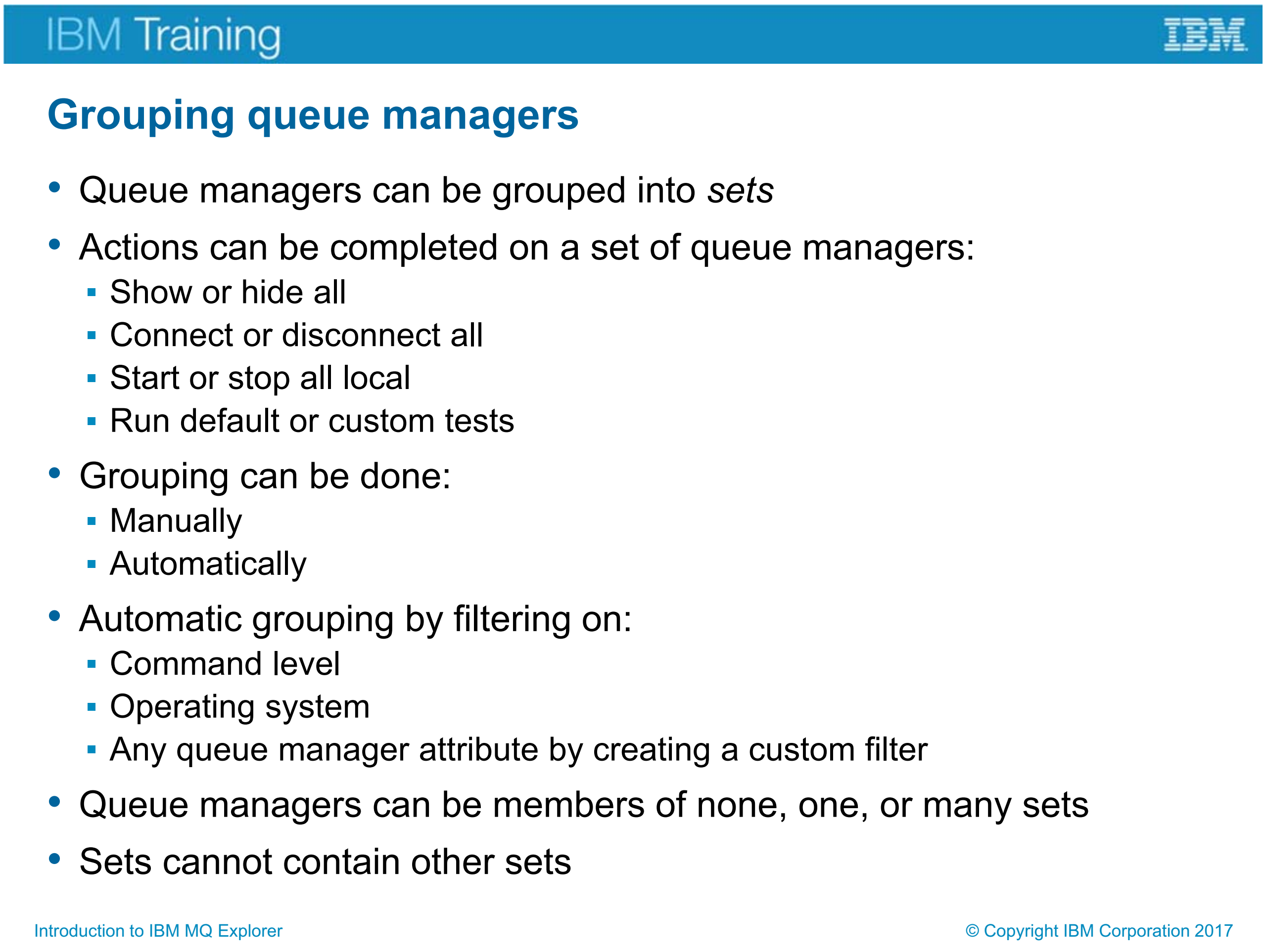
- General
- Extended
- Exits
- Cluster
- Repository
- Communication
- Events
- SSL
- Statistics
- Online monitoring
- Statistics monitoring
- Accounting monitoring
- Log
- XA resource manager
- Installable services
- Channels
- TCP
- LU6.2
- NetBIOS
- SPX
- Publish/subscribe

Introduction to IBM MQ Explorer

© Copyright IBM Corporation 2017

Figure 4-12. Queue manager property categories

This figure lists the queue manager property categories.



IBM Training

Grouping queue managers

- Queue managers can be grouped into *sets*
- Actions can be completed on a set of queue managers:
  - Show or hide all
  - Connect or disconnect all
  - Start or stop all local
  - Run default or custom tests
- Grouping can be done:
  - Manually
  - Automatically
- Automatic grouping by filtering on:
  - Command level
  - Operating system
  - Any queue manager attribute by creating a custom filter
- Queue managers can be members of none, one, or many sets
- Sets cannot contain other sets

Introduction to IBM MQ Explorer © Copyright IBM Corporation 2017

Figure 4-13. Grouping queue managers

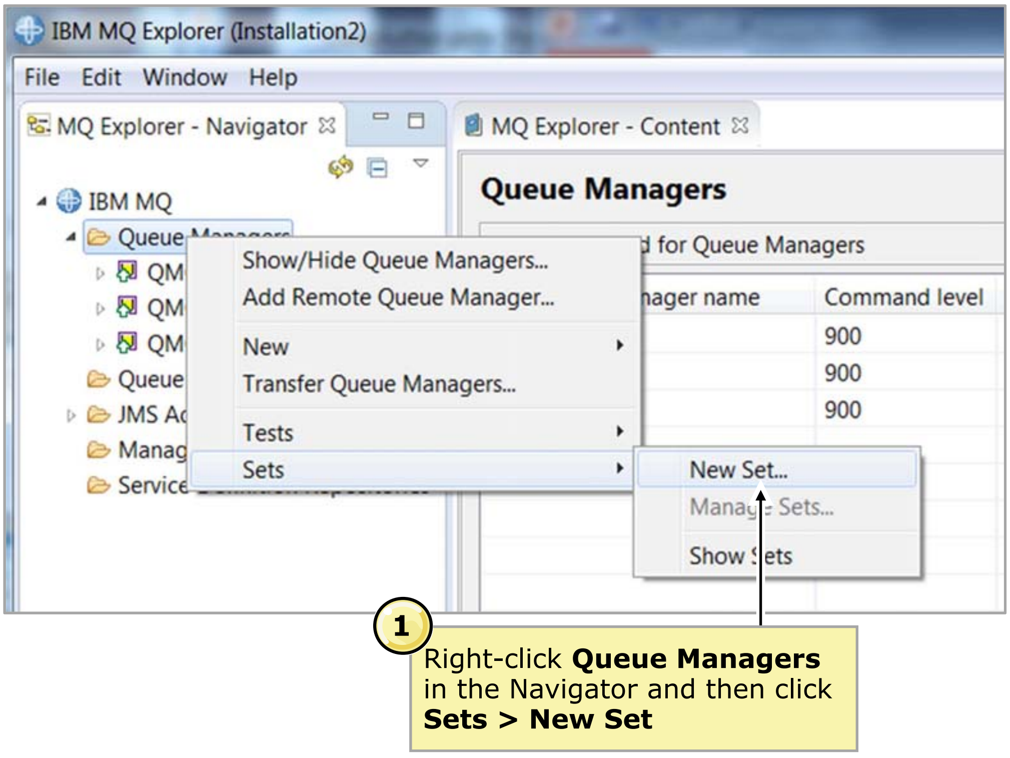
You can group queue managers into sets.

Grouping queue managers into sets makes it more efficient to complete operations on a select set of queue managers at the same time rather than one at a time. For example, you can stop and start all queue managers in a group at the same time.

You can group queue managers manually by selecting the queue managers. You can also group queue managers automatically by defining filter conditions. For example, you can define a group that includes running queue managers, or queue managers that use the same dead-letter queue.

IBM Training

Creating a queue manager set (1 of 7)



Right-click **Queue Managers** in the Navigator and then click **Sets > New Set**

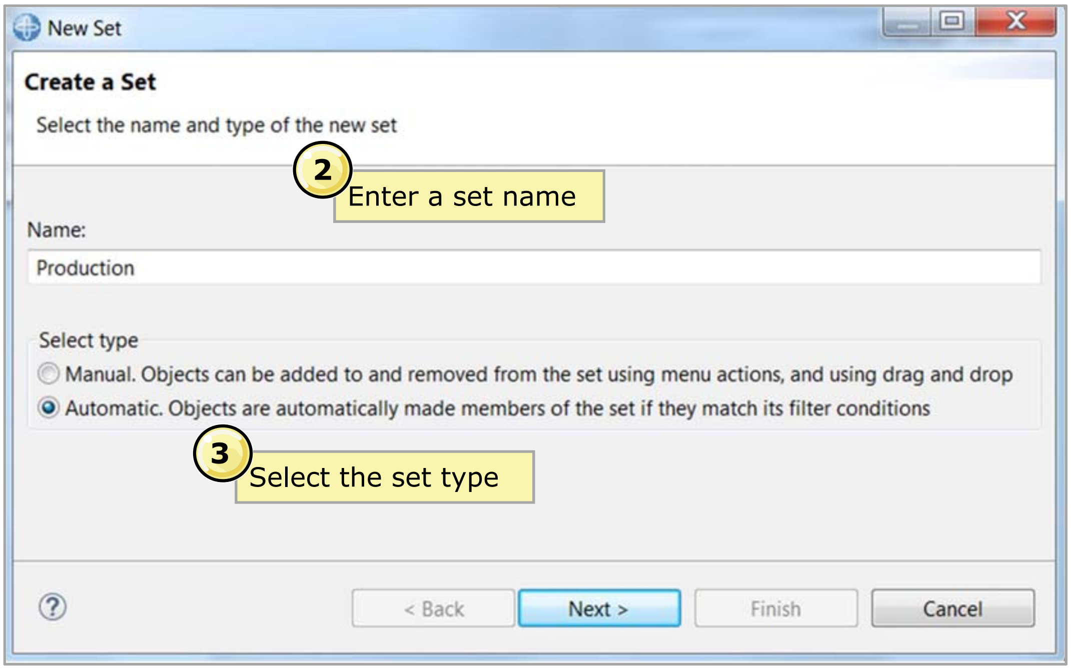
Figure 4-14. Creating a queue manager set (1 of 7)

This example shows you how to create an automatic queue manager that uses a custom filter to determine members of the group. In this example, the queue manager is automatically added to the group if the queue manager **Description** attribute is set to **`Production`**.

1. Right-click **Queue Managers** in the **MQ Explorer - Navigator** view and then click **Sets > New Set**.

IBM Training

Creating a queue manager set (2 of 7)



Create a Set

Select the name and type of the new set

Name: Production

Select type

( ) Manual. Objects can be added to and removed from the set using menu actions, and using drag and drop

(●) Automatic. Objects are automatically made members of the set if they match its filter conditions

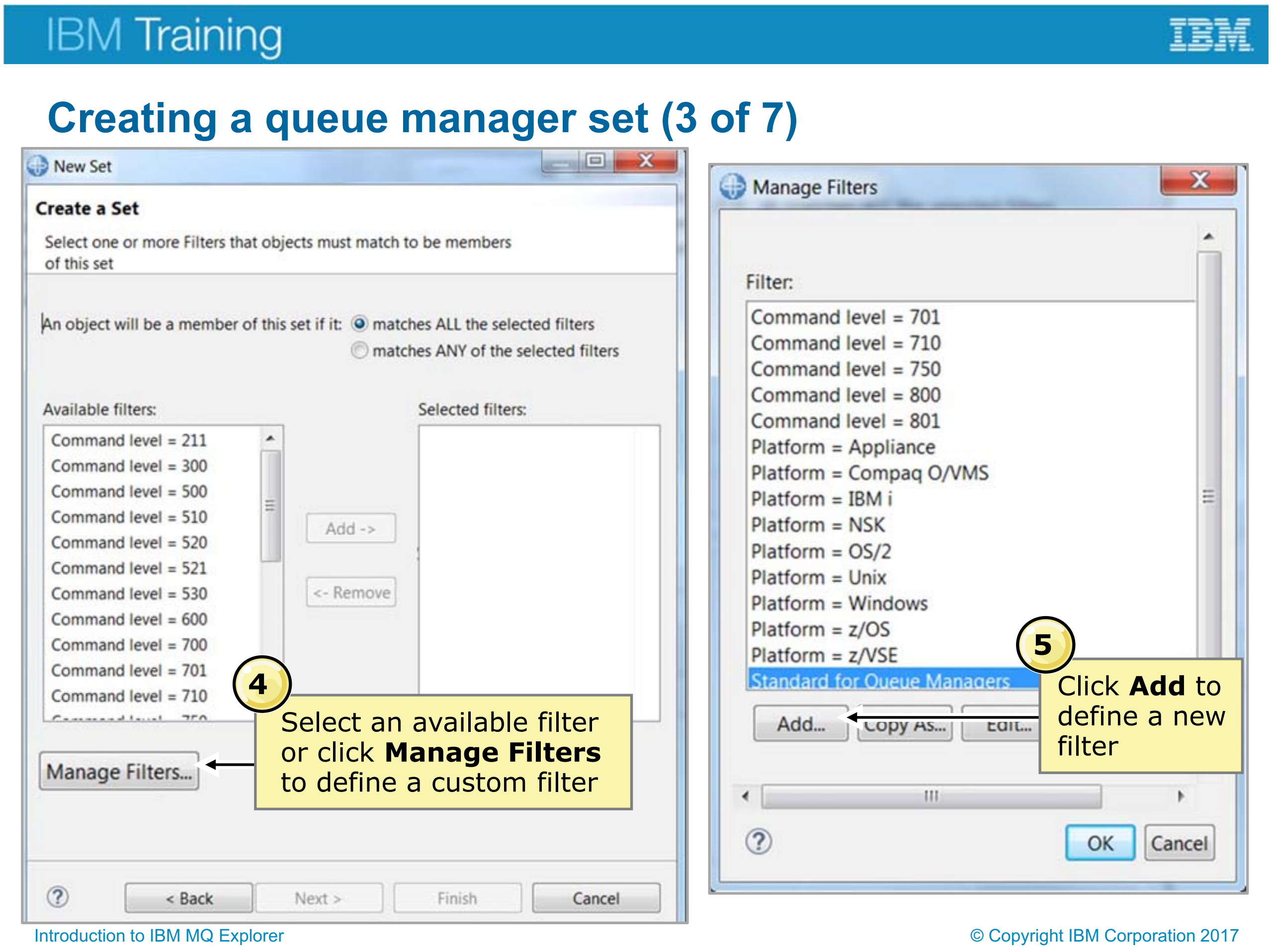
Introduction to IBM MQ Explorer

Figure 4-15. Creating a queue manager set (2 of 7)

2. Enter a descriptive set name.

   For example, create a set that is named **Production** that includes all queue managers that are identified as queue managers that are used in a production environment.

3. If you want to select the queue managers to add to the set by name, select **Manual**. If you want to use a filter to determine the queue managers that are added to the set, select **Automatic**. The example creates an automatic set that uses a custom filter.



Creating a queue manager set (3 of 7)

New Set

Create a Set

Select one or more Filters that objects must match to be members of this set

An object will be a member of this set if it: (●) matches ALL the selected filters ( ) matches ANY of the selected filters

Available filters:
Command level = 211
Command level = 300
Command level = 500
Command level = 510
Command level = 520
Command level = 521
Command level = 530
Command level = 600
Command level = 700
Command level = 701
Command level = 710

Selected filters:

4 Select an available filter or click **Manage Filters** to define a custom filter

Manage Filters

Filter:
Command level = 701
Command level = 710
Command level = 750
Command level = 800
Command level = 801
Platform = Appliance
Platform = Compaq O/VMS
Platform = IBM i
Platform = NSK
Platform = OS/2
Platform = Unix
Platform = Windows
Platform = z/OS
Platform = z/VSE
Standard for Queue Managers

5 Click **Add** to define a new filter

Introduction to IBM MQ Explorer © Copyright IBM Corporation 2017

Figure 4-16. Creating a queue manager set (3 of 7)

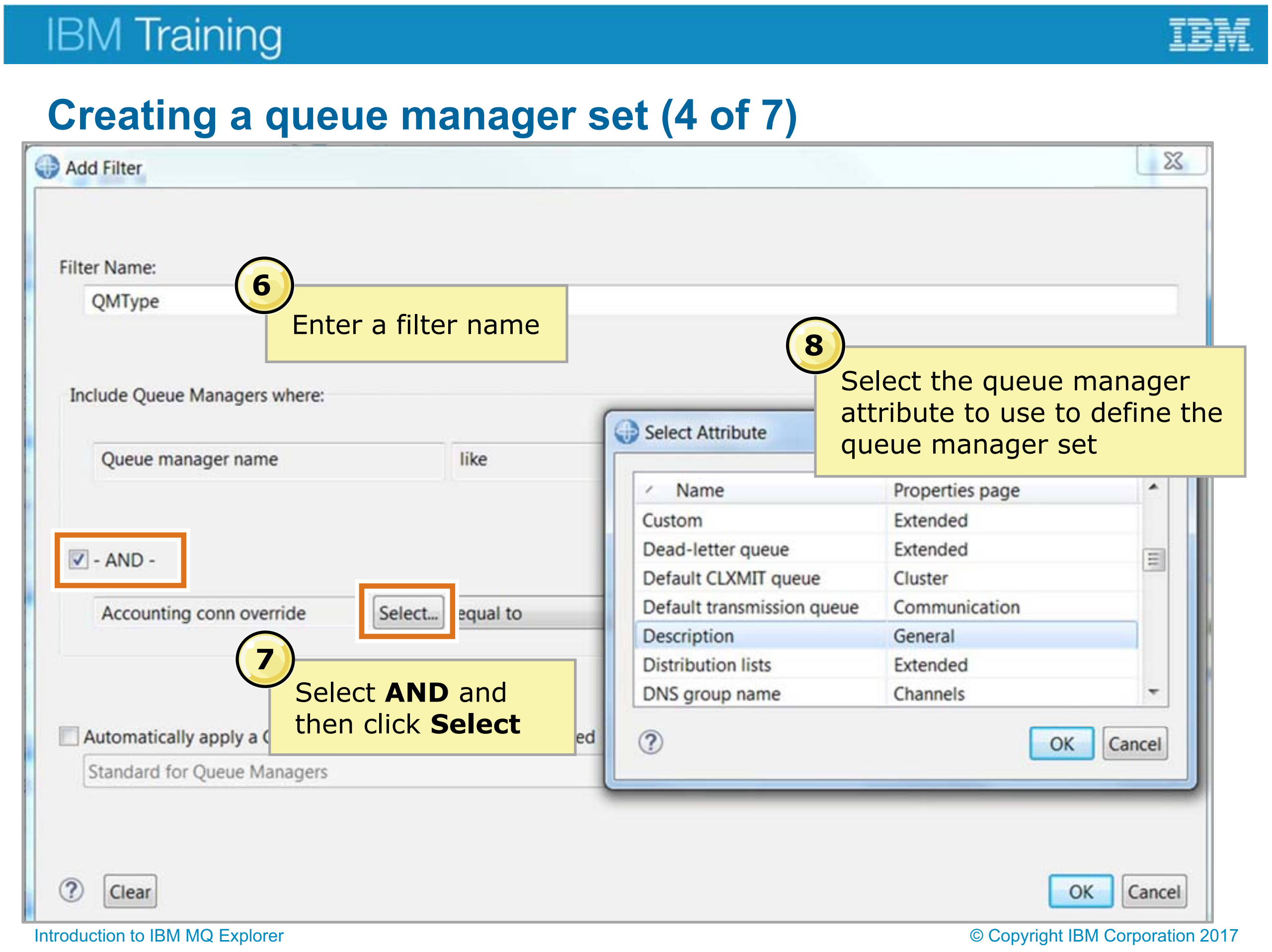
If you selected **Manual** as the set type, the final step for creating a queue manager set is to select the queue managers that are members of the set.

If you selected **Automatic** for the queue manager set type, the next steps are to:

4. Select the filter that automatically adds queue managers. For example, you can create a queue manager set that is named “Windows Queue Managers”. You can then select the filter condition of **Platform = Windows** so that any queue managers that are running on Windows are automatically added to the set.

   As an option, you can click **Manage Filters** to define a custom filter. The example in this figure and the next set of figures show the steps for defining a custom filter.

5. Click **Add** on the **Manage Filters** window to define a new filter.



Creating a queue manager set (4 of 7)

Add Filter
Filter Name:
QMType
6 Enter a filter name
Include Queue Managers where:
Queue manager name
like
- AND -
Accounting conn override
Select...
equal to
7 Select **AND** and then click **Select**
Automatically apply a
Standard for Queue Managers
Clear
OK
Cancel
Select Attribute
8 Select the queue manager attribute to use to define the queue manager set
OK
Cancel

Name	Properties page
Custom	Extended
Dead-letter queue	Extended
Default CLXMIT queue	Cluster
Default transmission queue	Communication
Description	General
Distribution lists	Extended
DNS group name	Channels

Introduction to IBM MQ Explorer
© Copyright IBM Corporation 2017

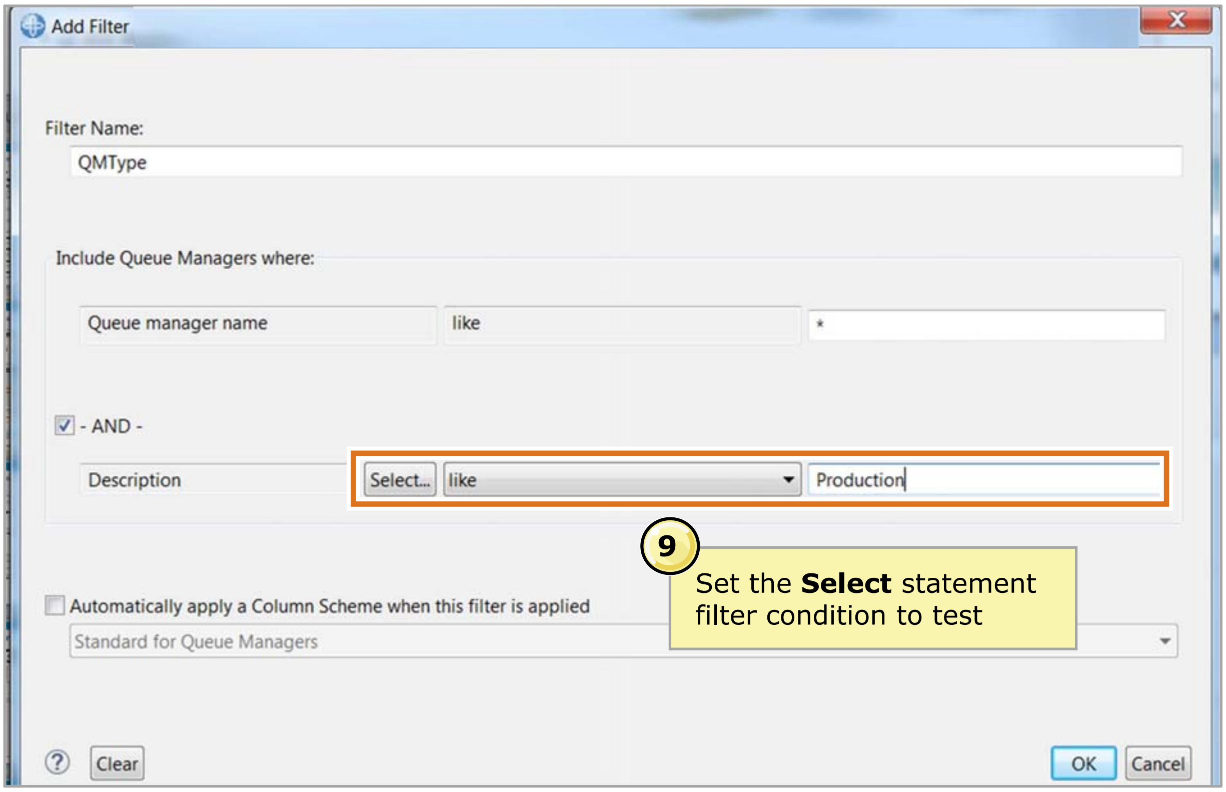
Figure 4-17. Creating a queue manager set (4 of 7)

This figure shows the next steps for creating a custom queue manager set filter for all queue managers.

6. Enter a unique descriptive filter name.
7. Select the **AND** check box and then click **Select** to continue to define the custom filter expression.
8. Select the queue manager attribute in the **Select Attribute** window and then click **OK**.

IBM Training

Creating a queue manager set (5 of 7)



Filter Name:
QMType

Include Queue Managers where:

Queue manager name | like | *

☑ - AND -

Description | like | Production

Set the **Select** statement filter condition to test

☐ Automatically apply a Column Scheme when this filter is applied
Standard for Queue Managers

Introduction to IBM MQ Explorer © Copyright IBM Corporation 2017

Figure 4-18. Creating a queue manager set (5 of 7)

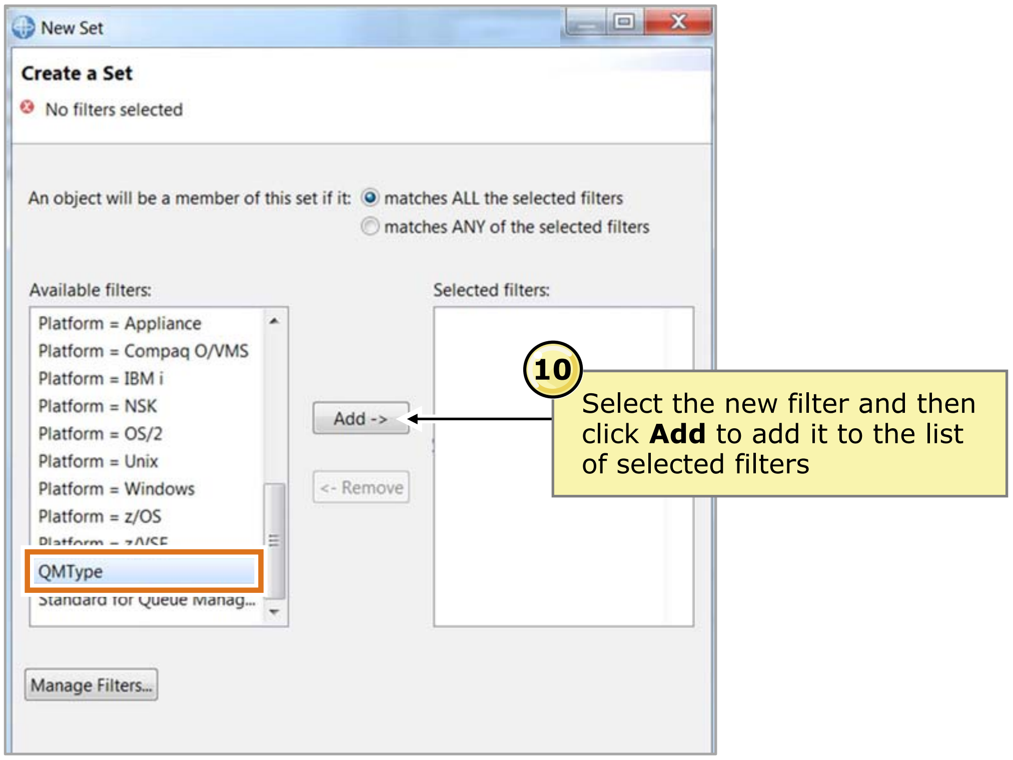
This figure shows the next step for defining a custom filter expression.

9. Select the expression condition such as **equal to** or **like** and then enter the string to evaluate.

   In this example, the queue manager is automatically added to the group if the queue manager **Description** attribute is set to **`Production`**.

IBM Training

Creating a queue manager set (6 of 7)


New Set

Create a Set

No filters selected

An object will be a member of this set if it: [✓ matches ALL the selected filters] matches ANY of the selected filters

Available filters:

- Platform = Appliance
- Platform = Compaq O/VMS
- Platform = IBM i
- Platform = NSK
- Platform = OS/2
- Platform = Unix
- Platform = Windows
- Platform = z/OS
- Platform = z/VSE
- QMType
- Standard for Queue Manag...

Selected filters:

Add ->
<- Remove
Manage Filters...
10
Select the new filter and then click **Add** to add it to the list of selected filters

Figure 4-19. Creating a queue manager set (6 of 7)

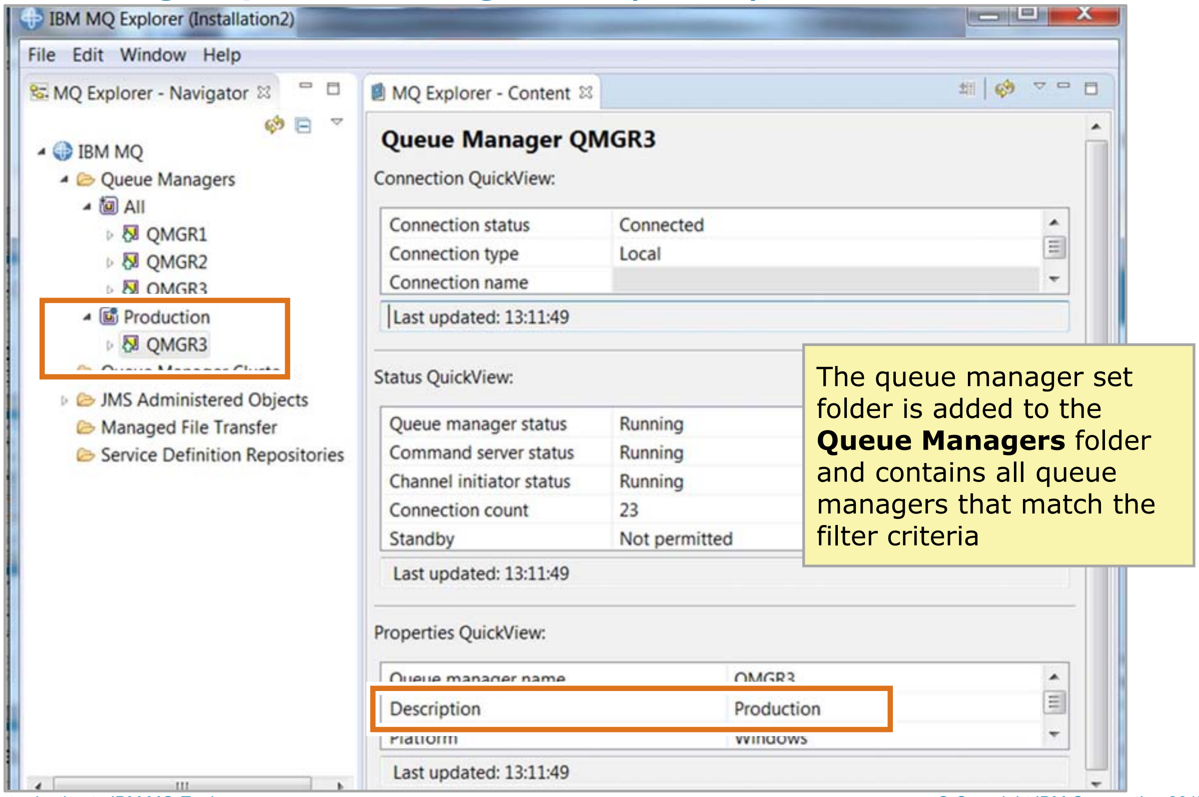
This figure shows the final step for defining an automatic queue manager set.

10. Select the filter in the **Available filters** pane and then click **Add** to move it to the **Selected** filters pane.

If you add more than one filter, be sure to verify the options for whether an object must match ALL selected filters or ANY of the selected filters.

IBM Training

Creating a queue manager set (7 of 7)



Queue Manager QMGR3

Connection QuickView:

Connection status	Connected
Connection type	Local
Connection name	

Last updated: 13:11:49

Status QuickView:

Queue manager status	Running
Command server status	Running
Channel initiator status	Running
Connection count	23
Standby	Not permitted

Last updated: 13:11:49

Properties QuickView:

Queue manager name	QMGR3
Description	Production
Platform	Windows

Last updated: 13:11:49

The queue manager set folder is added to the **Queue Managers** folder and contains all queue managers that match the filter criteria

Introduction to IBM MQ Explorer © Copyright IBM Corporation 2017

Figure 4-20. Creating a queue manager set (7 of 7)

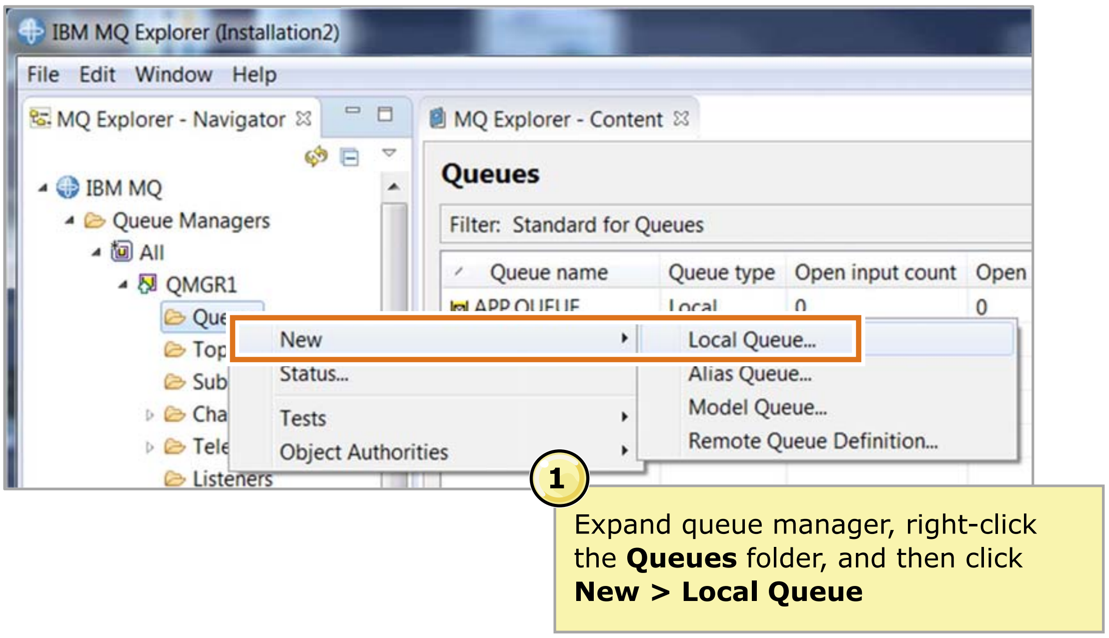
After you define a queue manager set, the **MQ Explorer- Navigator** view shows the new queue manager set and another queue manager set that is named **All**. The **All** set contains all queue managers that MQ Explorer can manage.

In this example, the queue manager that is named QMGR3 is automatically added to the **Production** set because the queue manager **Description** property is set to **`Production`**. You can create another queue manager set that is named **Development** and define another custom filter where **Description** is equal to **`Development`** to further separate and manage the queue managers by use.

You create a queue manager set in the exercise for this unit.

IBM Training

Creating a local queue (1 of 3)



Expand queue manager, right-click the **Queues** folder, and then click **New > Local Queue**

Introduction to IBM MQ Explorer © Copyright IBM Corporation 2017

Figure 4-21. Creating a local queue (1 of 3)

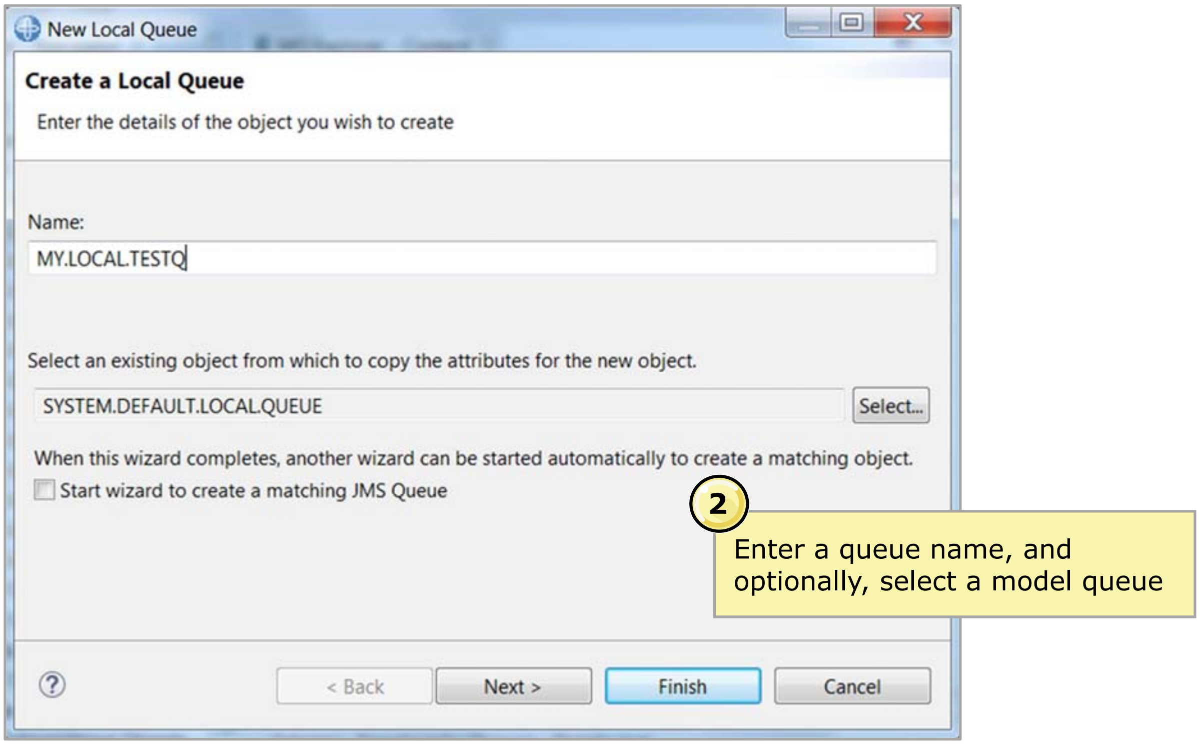
It is possible to define local, alias, model, and local definitions of remote queues for any queue manager that is connected to MQ Explorer.

To create a new local queue in MQ Explorer:

1. Right-click the **Queues** folder under the queue manager in the **MQ Explorer - Navigator** and then click **New > Local Queue**.

IBM Training

Creating a local queue (2 of 3)



New Local Queue

Create a Local Queue

Enter the details of the object you wish to create

Name:

MY.LOCAL.TESTQ

Select an existing object from which to copy the attributes for the new object.

SYSTEM.DEFAULT.LOCAL.QUEUE

When this wizard completes, another wizard can be started automatically to create a matching object.

☐ Start wizard to create a matching JMS Queue

Figure 4-22. Creating a local queue (2 of 3)

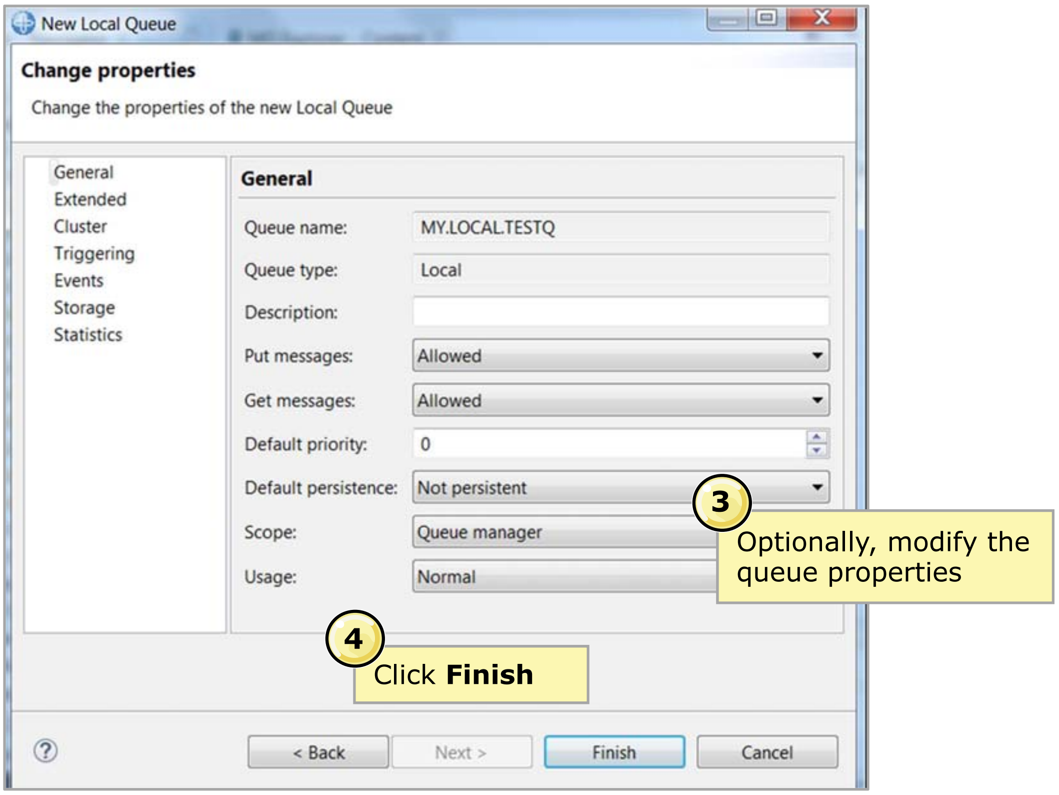
2. Enter a unique queue name.

   By default, MQ uses a default SYSTEM queue as the model for the new queue. If you want, you can specify a different queue to use as the model for this queue.

   An option also exists to automatically start another wizard to create a matching JMS queue.

IBM Training

Creating a local queue (3 of 3)


New Local Queue

Change properties

Change the properties of the new Local Queue

General

Queue name: MY.LOCAL.TESTQ

Queue type: Local

Description:

Put messages: Allowed

Get messages: Allowed

Default priority: 0

Default persistence: Not persistent

Scope: Queue manager

Usage: Normal

Introduction to IBM MQ Explorer © Copyright IBM Corporation 2017

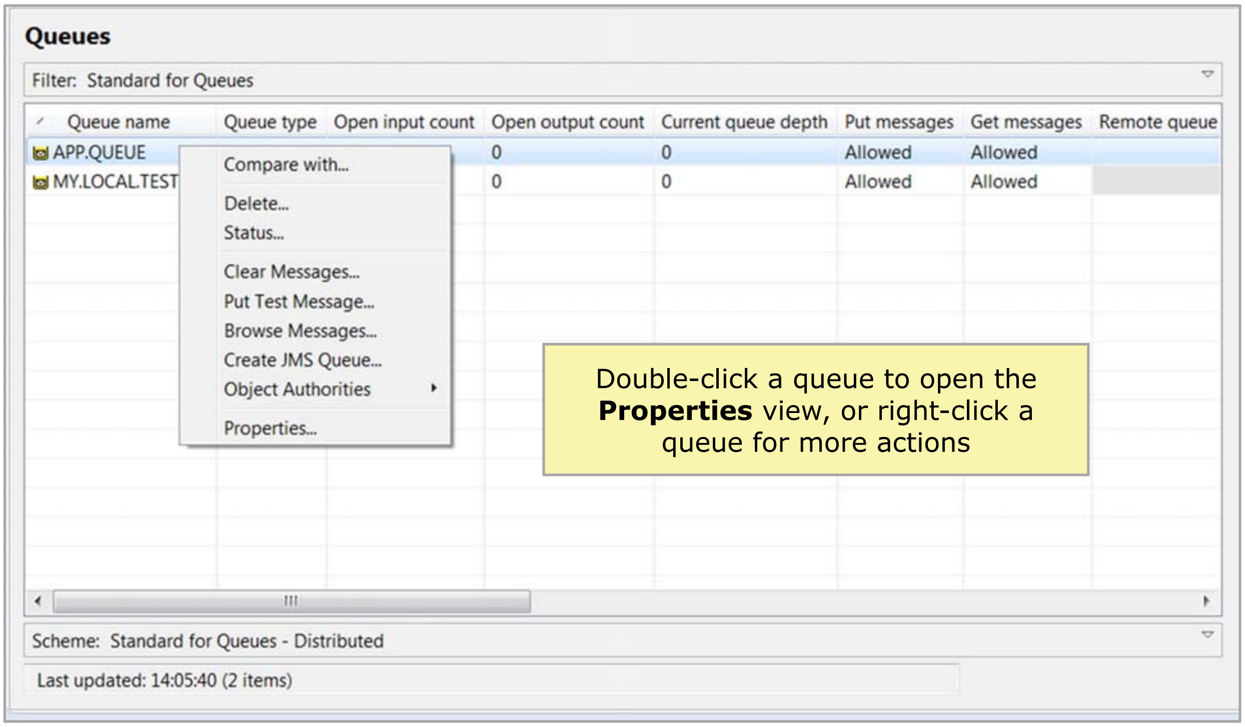
Figure 4-23. Creating a local queue (3 of 3)

This figure shows the next steps for creating a local queue.

3. Modify the queue properties, if necessary. The queue properties are categorized by function:
   - General
   - Extended
   - Cluster
   - Triggering
   - Events
   - Storage
   - Statistics
4. Click **Finish**.

IBM Training

Queues content view


Queues

Filter: Standard for Queues

Queue name	Queue type	Open input count	Open output count	Current queue depth	Put messages	Get messages	Remote queue
APP.QUEUE			0	0	Allowed	Allowed	
MY.LOCAL.TEST			0	0	Allowed	Allowed	

Double-click a queue to open the **Properties** view, or right-click a queue for more actions

Scheme: Standard for Queues - Distributed

Last updated: 14:05:40 (2 items)

Introduction to IBM MQ Explorer © Copyright IBM Corporation 2017

Figure 4-24. Queues content view

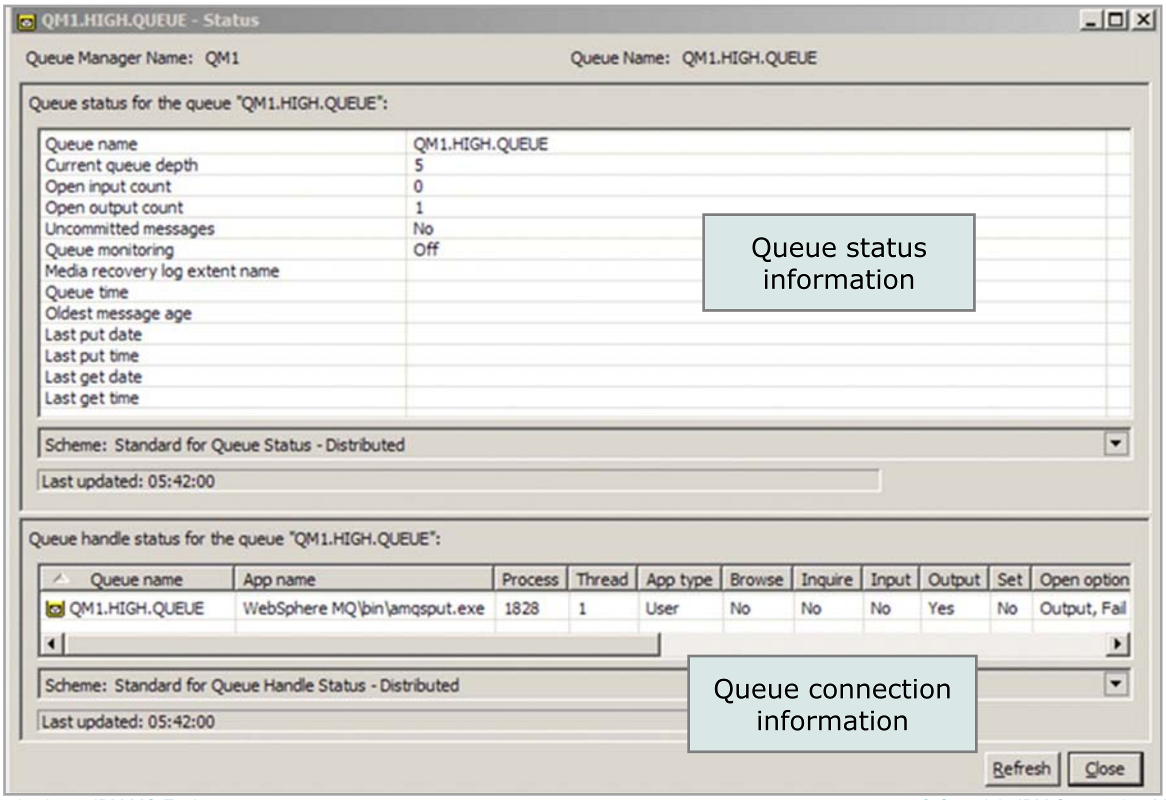
The **Queues** content view is used to view, clear, put messages, and obtain status information for the selected queue. It is also used to define object authorities on the queue that identify groups and users that can put and get messages.

You can modify queue properties by double-clicking a queue or right-clicking a queue and then clicking **Properties**.

Right-click the queue for actions such as clearing a queue, putting test messages, getting messages, and browsing messages.

IBM Training

Queue status information



QM1.HIGH.QUEUE - Status

Queue Manager Name: QM1 Queue Name: QM1.HIGH.QUEUE

Queue status for the queue "QM1.HIGH.QUEUE":

Queue name	QM1.HIGH.QUEUE
Current queue depth	5
Open input count	0
Open output count	1
Uncommitted messages	No
Queue monitoring	Off
Media recovery log extent name	
Queue time	
Oldest message age	
Last put date	
Last put time	
Last get date	
Last get time	

Scheme: Standard for Queue Status - Distributed

Last updated: 05:42:00

Queue handle status for the queue "QM1.HIGH.QUEUE":

Queue name	App name	Process	Thread	App type	Browse	Inquire	Input	Output	Set	Open option
QM1.HIGH.QUEUE	WebSphere MQ\bin\amqsput.exe	1828	1	User	No	No	No	Yes	No	Output, Fail

Scheme: Standard for Queue Handle Status - Distributed

Last updated: 05:42:00

Figure 4-25. Queue status information

The queue status view can help to determine real-time queue usage information. This information can assist application programmers with diagnostic information.

Obtain status information by right-clicking the queue in the **Queues** content view and then clicking **Status**.

Status information shows many of the same attributes that are returned by the **DISPLAY** command:

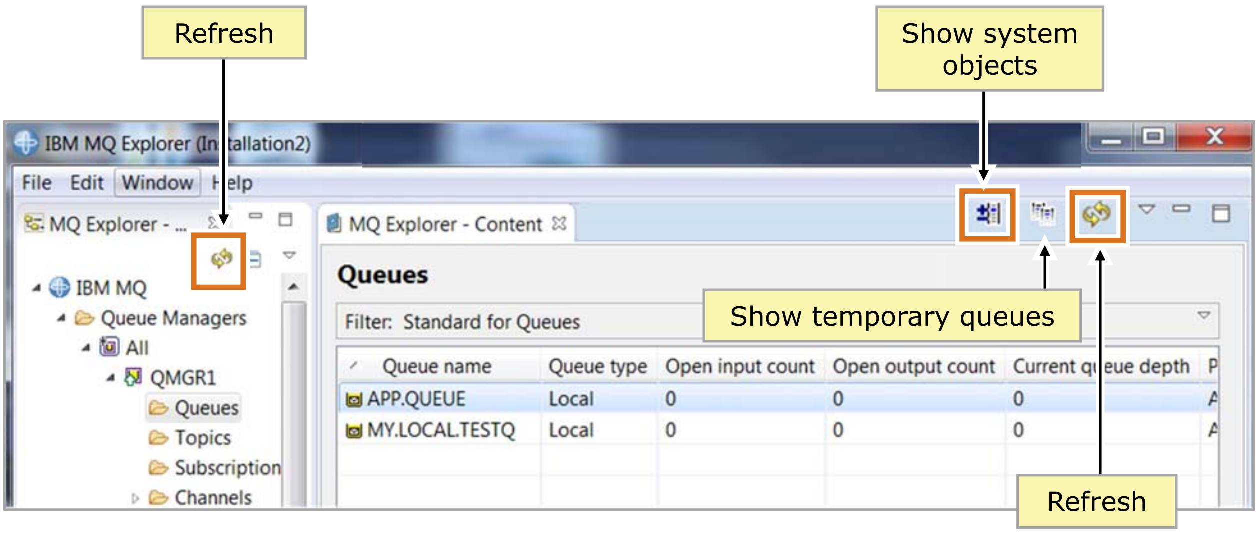
- Number of uncommitted messages
- Last put and get times
- Queue connection information
- Application usage information

The **Queue handle status** view, which is shown in the figure, contains the active applications that are using the queue. It contains the process ID, number of threads, and open options (**MQGET** or **MQPUT**). It also shows what open options are requested and the user ID that the process is running under. The queue connection information can be instrumental in assisting application testing.

The **Queue handle status** example in the figure shows that an **amqsput** application with process number 1828 is using one thread.

IBM Training

Shortcut icons



Queue name	Queue type	Open input count	Open output count	Current queue depth
APP.QUEUE	Local	0	0	0
MY.LOCAL.TESTQ	Local	0	0	0

Introduction to IBM MQ Explorer

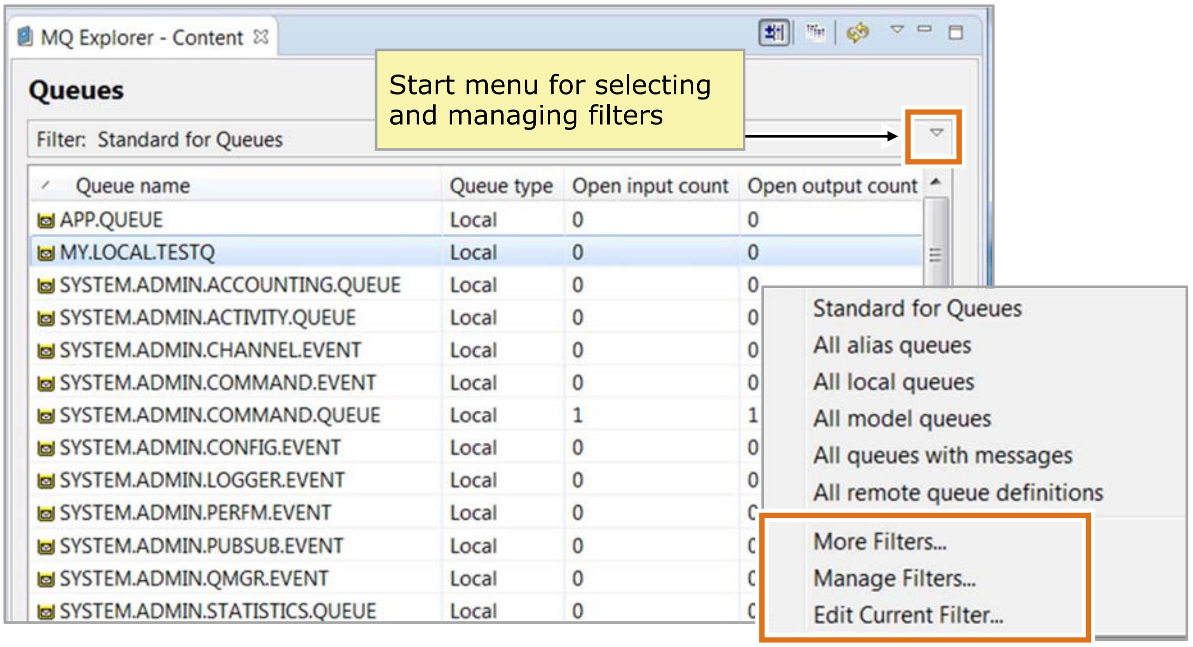
Figure 4-26. Shortcut icons

As shown in the figure, MQ Explorer has shortcut icons for refreshing lists and views and showing system objects.

More shortcut icons are available based on the selected content. For example, when the content area contains the list of queues, a shortcut icon is available for showing temporary queues and SYSTEM queues.

IBM Training

Viewing queue filters



MQ Explorer - Content

Queues

Filter: Standard for Queues

Queue name	Queue type	Open input count	Open output count
APP.QUEUE	Local	0	0
MY.LOCAL.TESTQ	Local	0	0
SYSTEM.ADMIN.ACCOUNTING.QUEUE	Local	0	0
SYSTEM.ADMIN.ACTIVITY.QUEUE	Local	0	0
SYSTEM.ADMIN.CHANNEL.EVENT	Local	0	0
SYSTEM.ADMIN.COMMAND.EVENT	Local	0	0
SYSTEM.ADMIN.COMMAND.QUEUE	Local	1	1
SYSTEM.ADMIN.CONFIG.EVENT	Local	0	0
SYSTEM.ADMIN.LOGGER.EVENT	Local	0	0
SYSTEM.ADMIN.PERFM.EVENT	Local	0	0
SYSTEM.ADMIN.PUBSUB.EVENT	Local	0	0
SYSTEM.ADMIN.QMGR.EVENT	Local	0	0
SYSTEM.ADMIN.STATISTICS.QUEUE	Local	0	0

Figure 4-27. Viewing queue filters

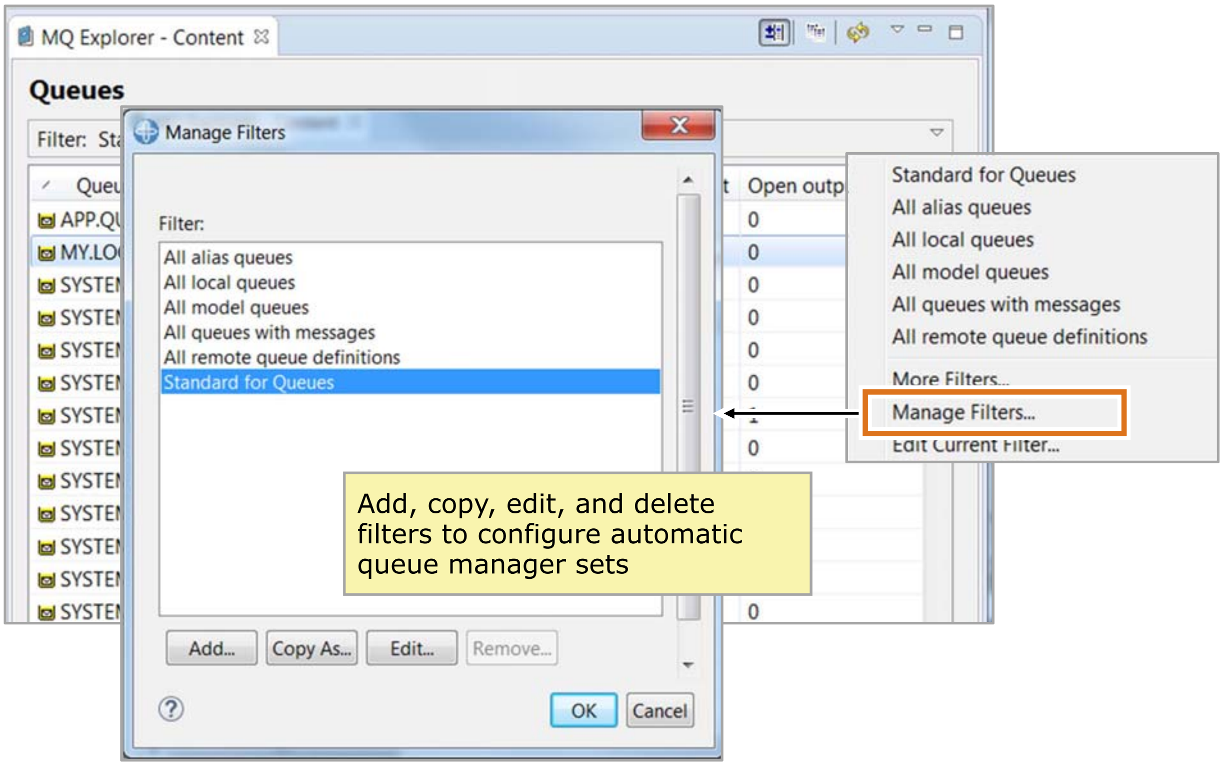
Filters provide a way to selectively view queues that meet certain criteria such as queues that contain messages. A filter allows the display of queues that meet the filter criteria and hides queues that do not conform to the selection criteria.

Click the **Filter** menu icon to access the options for selecting and managing the filters.

The **Show system objects** icon is a toggle. When active, it shows the system queues for the queue manager.

IBM Training

Managing queue filters



Add, copy, edit, and delete filters to configure automatic queue manager sets

Figure 4-28. Managing queue filters

By using the **Manage Filters** option, you can create, modify, and delete queue display filters.

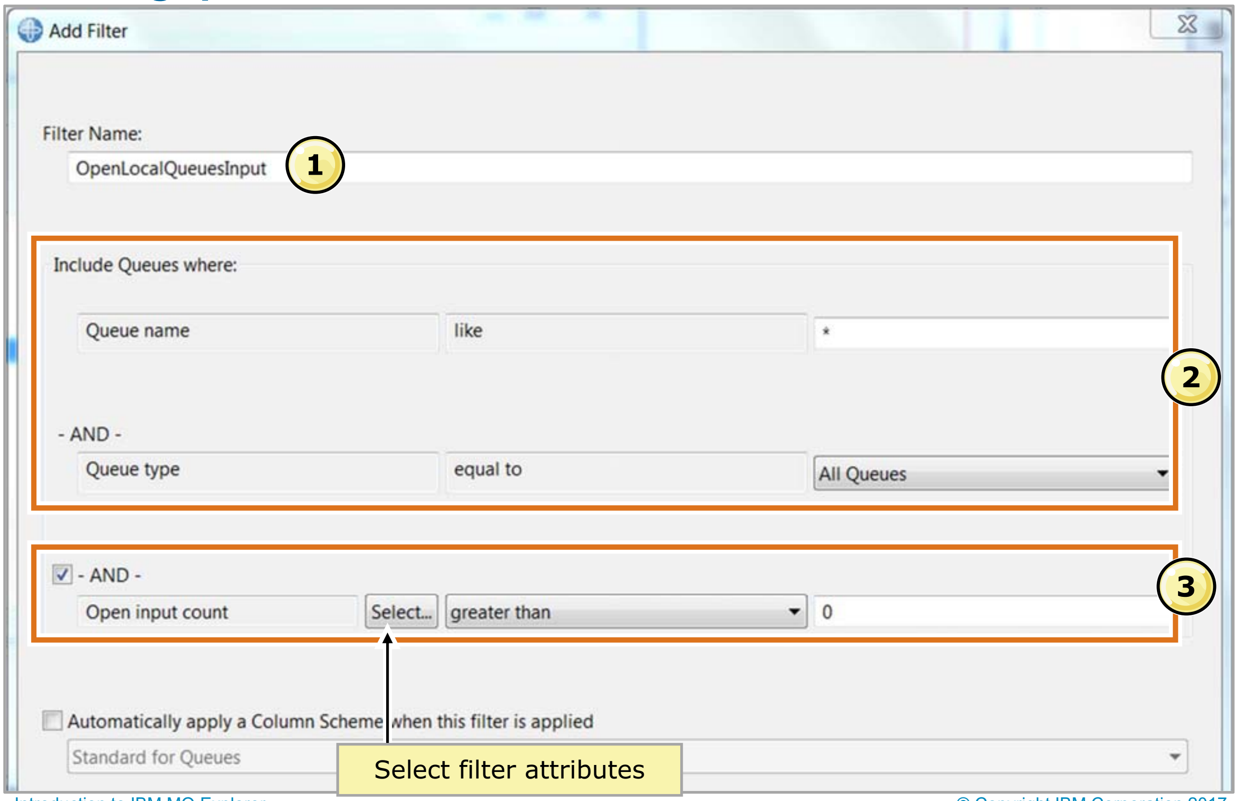
In the figure, the list of filters includes a filter to show all alias queues, all local queues, all model queues, all queues with messages, and all remote queues.

In the **Manage Filters** window, you can:

- Click **Add** to create a filter.
- Click **Copy As** to copy a filter.
- Click **Edit** to edit filters to add, remove, or change the criteria that are set for the filter.
- Click **Remove** to delete a filter.

IBM Training

Adding queue filters


Add Filter

Filter Name: OpenLocalQueuesInput

Include Queues where:

Queue name like *

- AND -

Queue type equal to All Queues

☑ - AND -

Open input count Select... greater than 0

☐ Automatically apply a Column Scheme when this filter is applied

Standard for Queues

Introduction to IBM MQ Explorer © Copyright IBM Corporation 2017

Figure 4-29. Adding queue filters

Using advanced filtering, you can customize the views of information gathering.

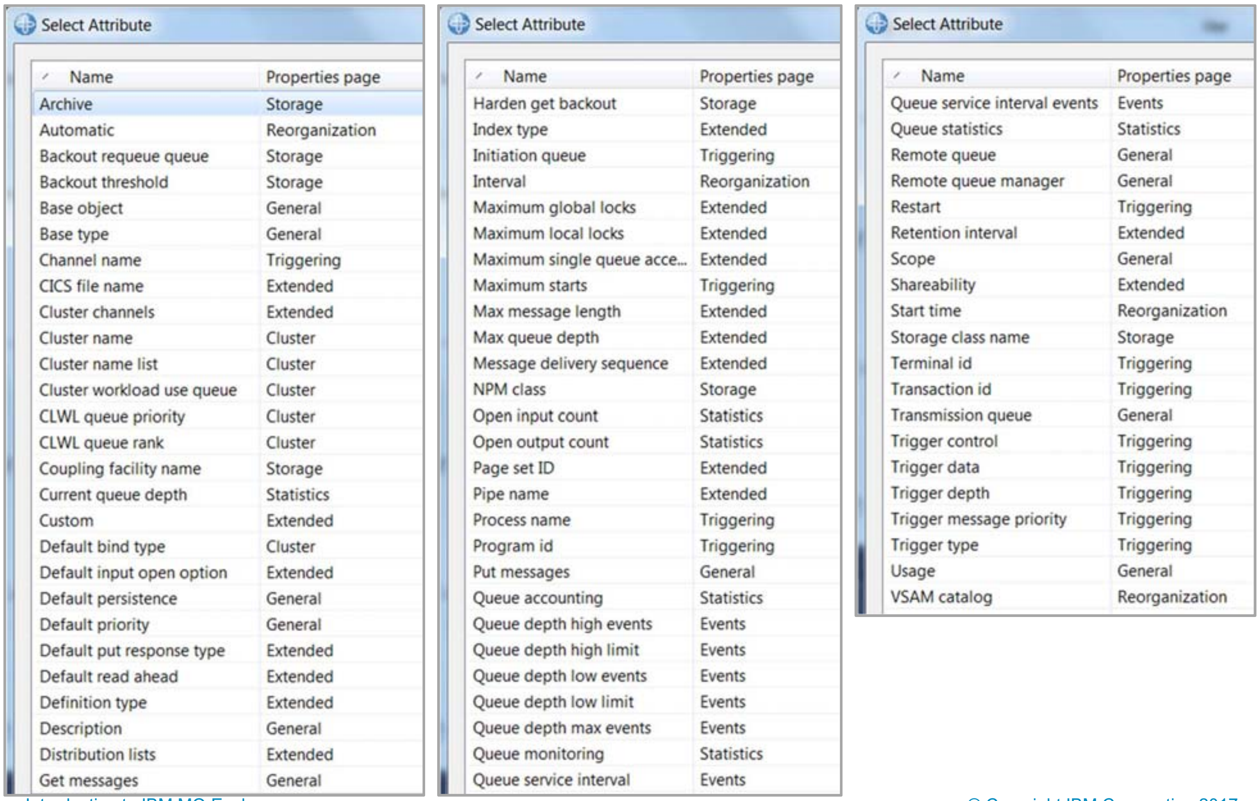
To add a queue display filter in MQ Explorer, click **Add** on the **Manage filters** window. The figure shows the next steps for defining a custom filter.

1. Enter a descriptive filter name.
2. Select the filter conditions.
3. Optionally, add a condition by selecting **AND** and the filter attributes. Click **Select** to select the filter attribute from a list of valid attributes.

In the example, a filter that is named **OpenLocalQueuesInput** filters all queues. The filter expression displays the queues where the open input count for the queue is not equal to zero.

IBM Training

Queue filter attributes



Select Attribute

Name	Properties page
Archive	Storage
Automatic	Reorganization
Backout requeue queue	Storage
Backout threshold	Storage
Base object	General
Base type	General
Channel name	Triggering
CICS file name	Extended
Cluster channels	Extended
Cluster name	Cluster
Cluster name list	Cluster
Cluster workload use queue	Cluster
CLWL queue priority	Cluster
CLWL queue rank	Cluster
Coupling facility name	Storage
Current queue depth	Statistics
Custom	Extended
Default bind type	Cluster
Default input open option	Extended
Default persistence	General
Default priority	General
Default put response type	Extended
Default read ahead	Extended
Definition type	Extended
Description	General
Distribution lists	Extended
Get messages	General
Harden get backout	Storage
Index type	Extended
Initiation queue	Triggering
Interval	Reorganization
Maximum global locks	Extended
Maximum local locks	Extended
Maximum single queue acce...	Extended
Maximum starts	Triggering
Max message length	Extended
Max queue depth	Extended
Message delivery sequence	Extended
NPM class	Storage
Open input count	Statistics
Open output count	Statistics
Page set ID	Extended
Pipe name	Extended
Process name	Triggering
Program id	Triggering
Put messages	General
Queue accounting	Statistics
Queue depth high events	Events
Queue depth high limit	Events
Queue depth low events	Events
Queue depth low limit	Events
Queue depth max events	Events
Queue monitoring	Statistics
Queue service interval	Events
Queue service interval events	Events
Queue statistics	Statistics
Remote queue	General
Remote queue manager	General
Restart	Triggering
Retention interval	Extended
Scope	General
Shareability	Extended
Start time	Reorganization
Storage class name	Storage
Terminal id	Triggering
Transaction id	Triggering
Transmission queue	General
Trigger control	Triggering
Trigger data	Triggering
Trigger depth	Triggering
Trigger message priority	Triggering
Trigger type	Triggering
Usage	General
VSAM catalog	Reorganization

Introduction to IBM MQ Explorer

© Copyright IBM Corporation 2017

Figure 4-30. Queue filter attributes

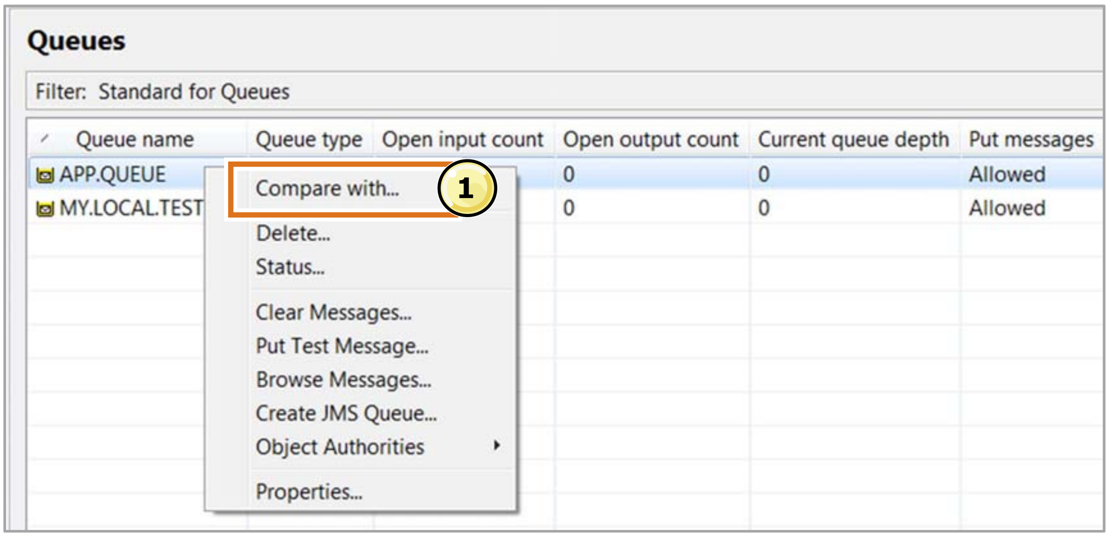
The figure shows the filter attributes that are available when you specify an extra **AND** condition on the queue filter expression. Examples of filter attributes are cluster name, current queue depth, maximum queue depth, and process name.

The filter attributes are displayed when you click **Select** in the **Add filter** view.

IBM Training

Comparing queues (1 of 2)

- Can compare two like resources
- Can compare resources across queue managers



Queue name	Queue type	Open input count	Open output count	Current queue depth	Put messages
APP.QUEUE			0	0	Allowed
MY.LOCAL.TEST			0	0	Allowed

Introduction to IBM MQ Explorer © Copyright IBM Corporation 2017

Figure 4-31. Comparing queues (1 of 2)

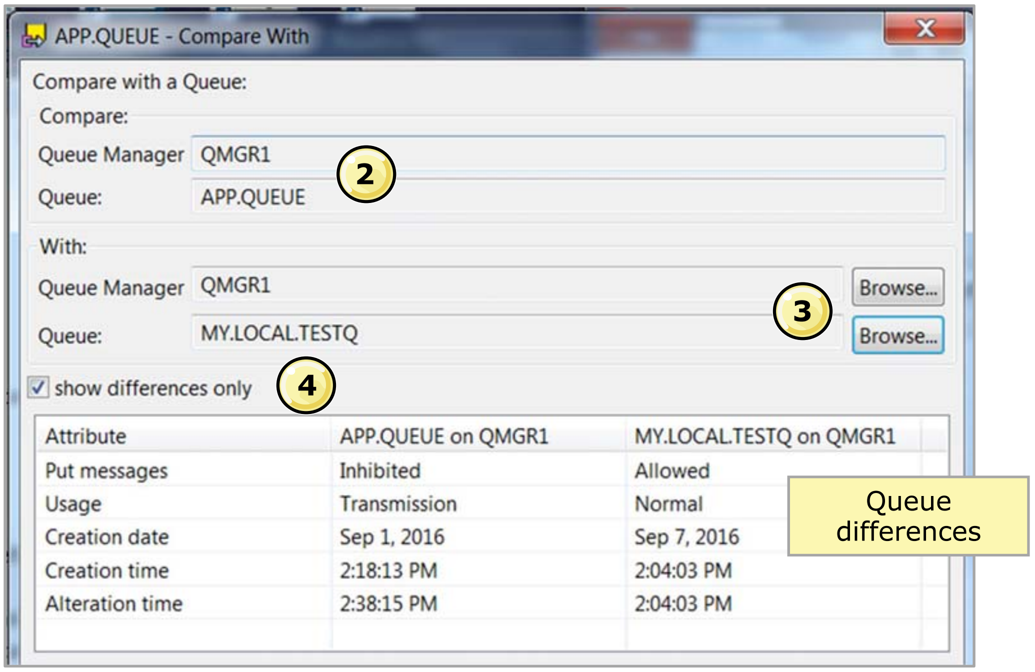
Using MQ Explorer, you can compare queues. The queue compare functions can be used as a migration aid to verify queue attributes after they are moved into production, for example.

To compare two queues, complete the following steps:

1. In the **Queues** view, right-click the first queue and then click **Compare with**.

IBM Training

Comparing queues (2 of 2)



Compare with a Queue:

Compare:

Queue Manager	QMGR1
Queue:	APP.QUEUE

With:

Queue Manager	QMGR1
Queue:	MY.LOCAL.TESTQ

☑ show differences only

Attribute	APP.QUEUE on QMGR1	MY.LOCAL.TESTQ on QMGR1
Put messages	Inhibited	Allowed
Usage	Transmission	Normal
Creation date	Sep 1, 2016	Sep 7, 2016
Creation time	2:18:13 PM	2:04:03 PM
Alteration time	2:38:15 PM	2:04:03 PM

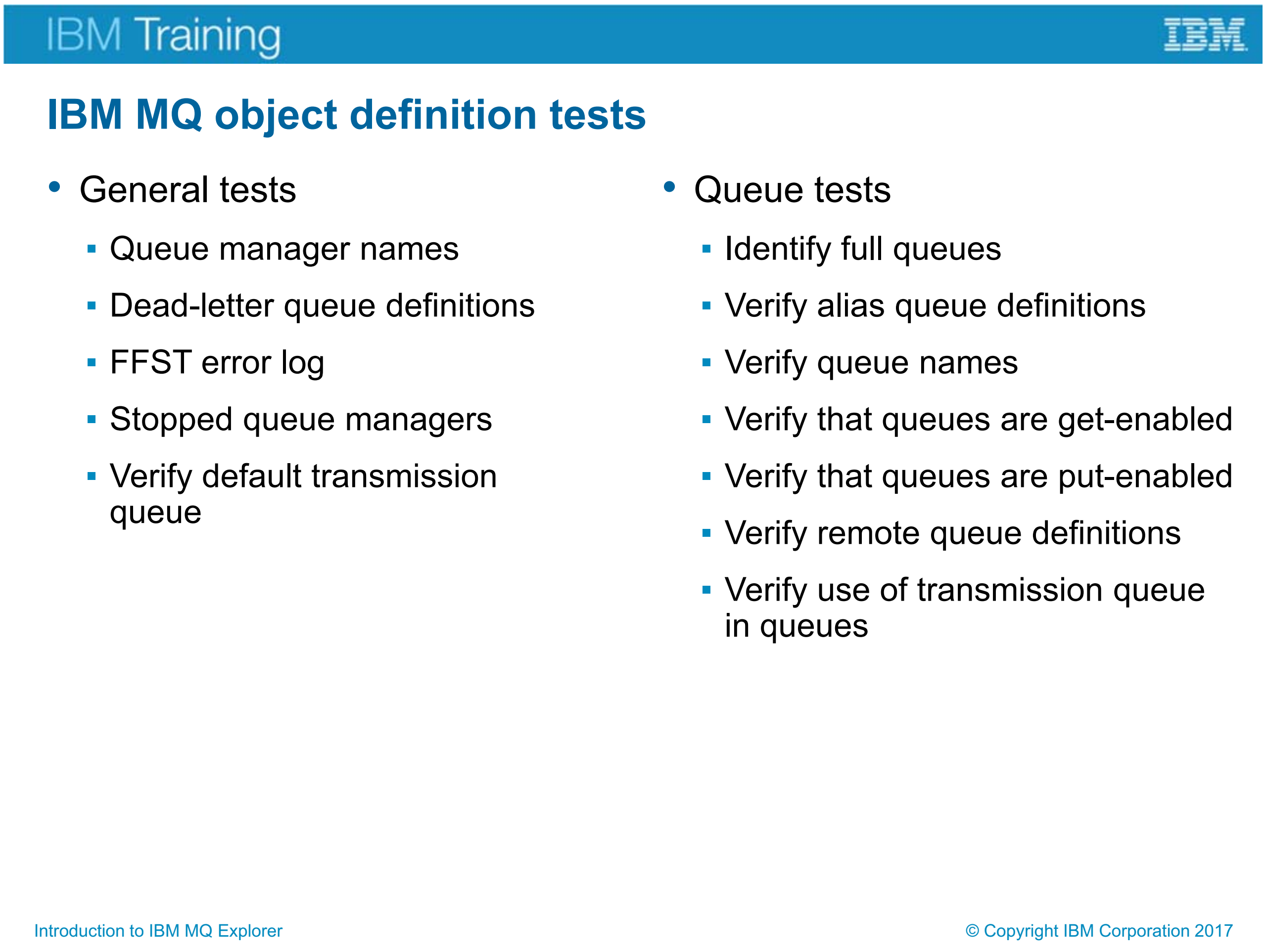
Figure 4-32. Comparing queues (2 of 2)

2. Select the queue manager for the second queue.
3. Select the queue in the **With** field.
4. Optionally, select **show differences only** to display the queue properties that are different.

   If you do not select **show differences only**, the list shows all the properties for both queues.

The example in the figure compares two queues from the same queue manager.

This example is previewing queue attributes that are different.



IBM Training

IBM MQ object definition tests

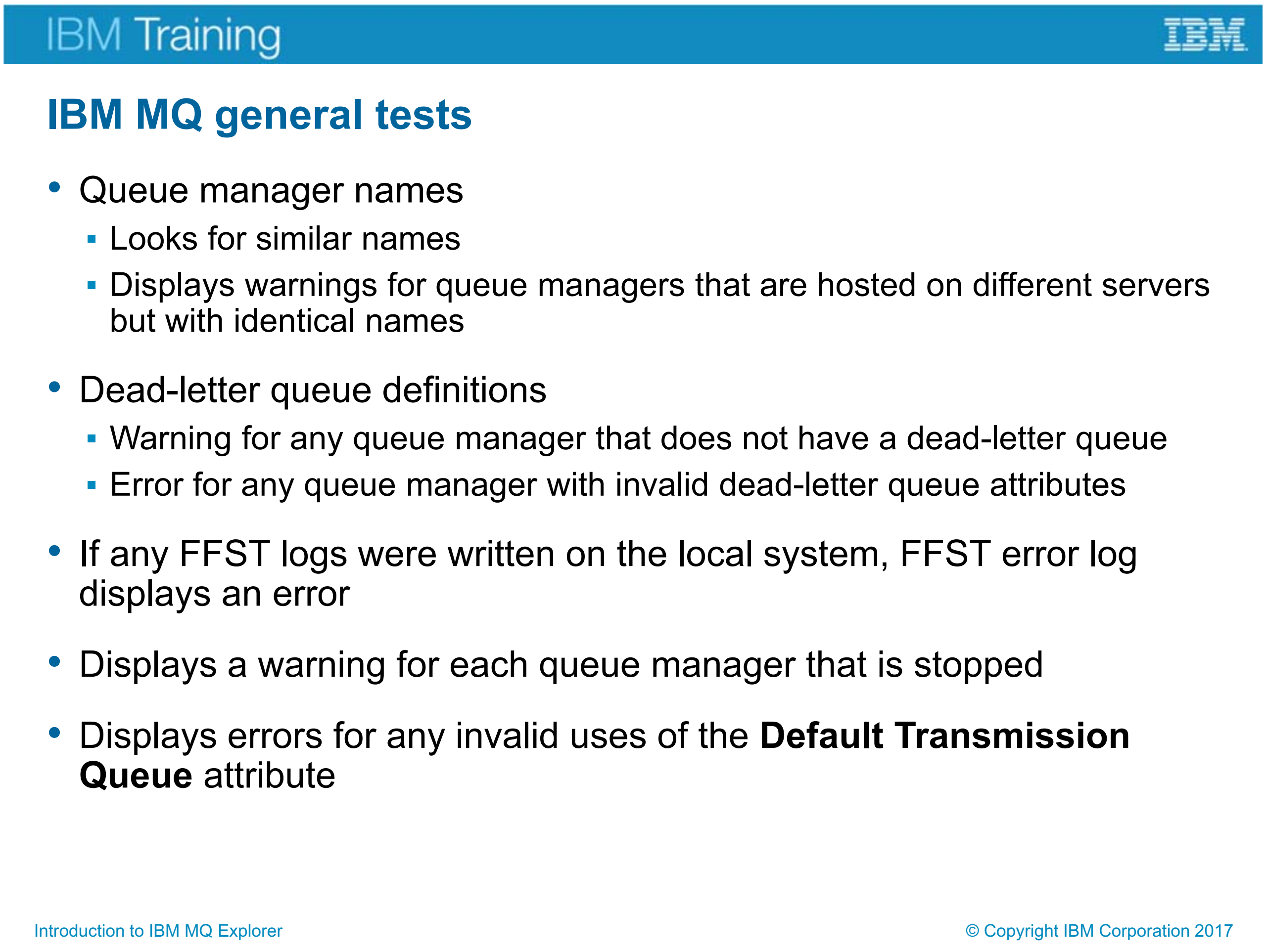
- General tests
  - Queue manager names
  - Dead-letter queue definitions
  - FFST error log
  - Stopped queue managers
  - Verify default transmission queue
- Queue tests
  - Identify full queues
  - Verify alias queue definitions
  - Verify queue names
  - Verify that queues are get-enabled
  - Verify that queues are put-enabled
  - Verify remote queue definitions
  - Verify use of transmission queue in queues

Introduction to IBM MQ Explorer © Copyright IBM Corporation 2017

Figure 4-33. IBM MQ object definition tests

You can use MQ Explorer to test the general functions of MQ, test queues, and test queue connectivity.

This figure summarizes the MQ Explorer object definition tests.



IBM Training

IBM MQ general tests

- Queue manager names
  - Looks for similar names
  - Displays warnings for queue managers that are hosted on different servers but with identical names
- Dead-letter queue definitions
  - Warning for any queue manager that does not have a dead-letter queue
  - Error for any queue manager with invalid dead-letter queue attributes
- If any FFST logs were written on the local system, FFST error log displays an error
- Displays a warning for each queue manager that is stopped
- Displays errors for any invalid uses of the **Default Transmission Queue** attribute

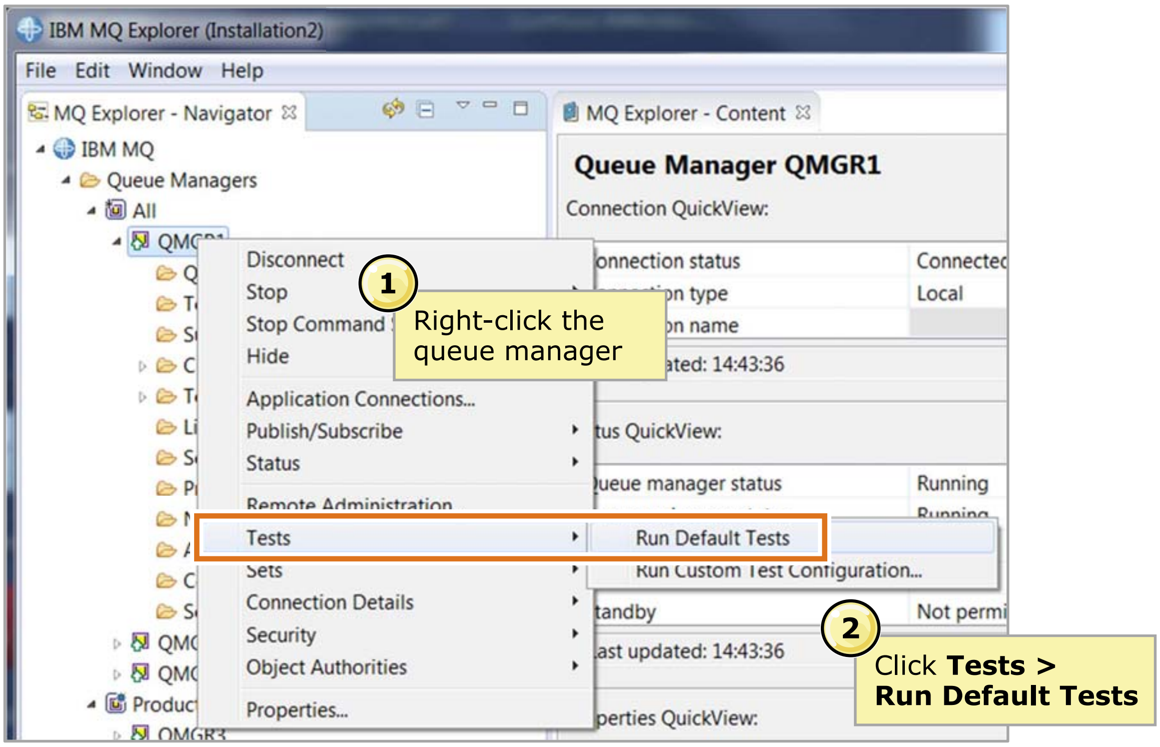
Introduction to IBM MQ Explorer © Copyright IBM Corporation 2017

Figure 4-34. IBM MQ general tests

The MQ Explorer tests provide some basic information about queue manager names, dead-letter queue definitions, the First Failure Support Technology (FFST) error log, stopped queue managers, and default transmission queues.

IBM Training

Running the IBM MQ tests



Introduction to IBM MQ Explorer © Copyright IBM Corporation 2017

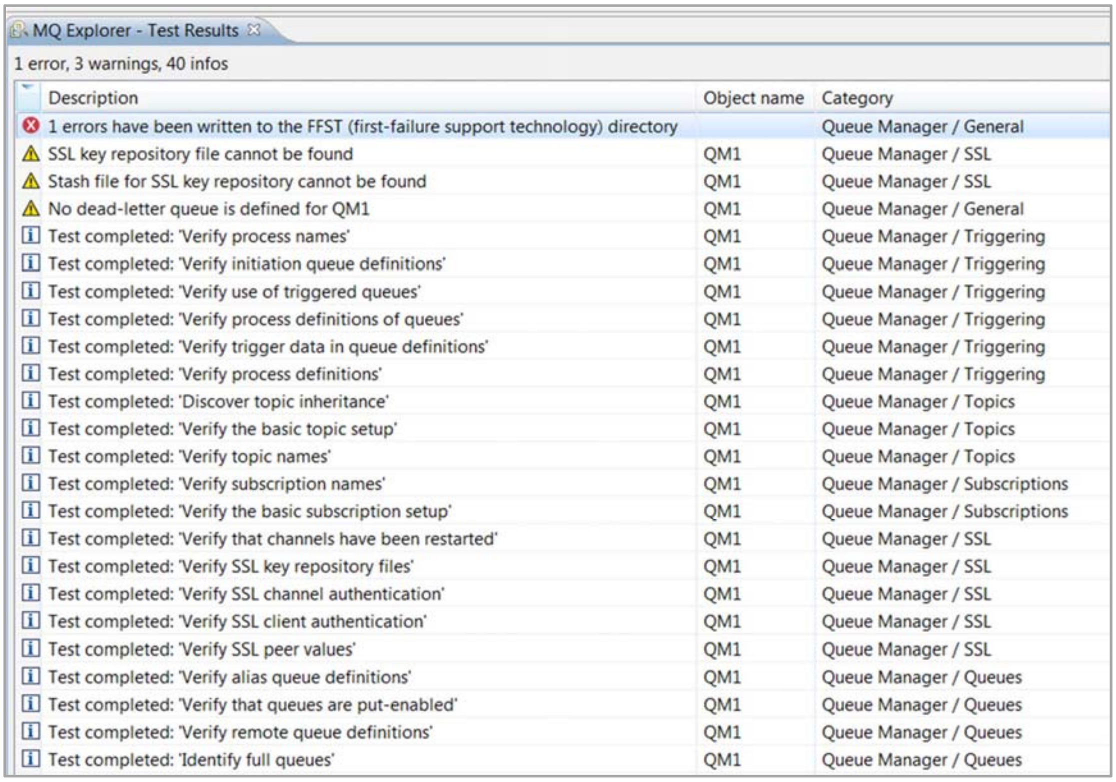
Figure 4-35. Running the IBM MQ tests

To run the MQ tests in MQ Explorer:

1. Right-click a queue manager in the **MQ Explorer - Navigator** view.
2. Click **Tests > Run Default Tests**.

IBM Training

IBM MQ test results example



MQ Explorer - Test Results

1 error, 3 warnings, 40 infos

Description	Object name	Category
1 errors have been written to the FFST (first-failure support technology) directory		Queue Manager / General
SSL key repository file cannot be found	QM1	Queue Manager / SSL
Stash file for SSL key repository cannot be found	QM1	Queue Manager / SSL
No dead-letter queue is defined for QM1	QM1	Queue Manager / General
Test completed: 'Verify process names'	QM1	Queue Manager / Triggering
Test completed: 'Verify initiation queue definitions'	QM1	Queue Manager / Triggering
Test completed: 'Verify use of triggered queues'	QM1	Queue Manager / Triggering
Test completed: 'Verify process definitions of queues'	QM1	Queue Manager / Triggering
Test completed: 'Verify trigger data in queue definitions'	QM1	Queue Manager / Triggering
Test completed: 'Verify process definitions'	QM1	Queue Manager / Triggering
Test completed: 'Discover topic inheritance'	QM1	Queue Manager / Topics
Test completed: 'Verify the basic topic setup'	QM1	Queue Manager / Topics
Test completed: 'Verify topic names'	QM1	Queue Manager / Topics
Test completed: 'Verify subscription names'	QM1	Queue Manager / Subscriptions
Test completed: 'Verify the basic subscription setup'	QM1	Queue Manager / Subscriptions
Test completed: 'Verify that channels have been restarted'	QM1	Queue Manager / SSL
Test completed: 'Verify SSL key repository files'	QM1	Queue Manager / SSL
Test completed: 'Verify SSL channel authentication'	QM1	Queue Manager / SSL
Test completed: 'Verify SSL client authentication'	QM1	Queue Manager / SSL
Test completed: 'Verify SSL peer values'	QM1	Queue Manager / SSL
Test completed: 'Verify alias queue definitions'	QM1	Queue Manager / Queues
Test completed: 'Verify that queues are put-enabled'	QM1	Queue Manager / Queues
Test completed: 'Verify remote queue definitions'	QM1	Queue Manager / Queues
Test completed: 'Identify full queues'	QM1	Queue Manager / Queues

Introduction to IBM MQ Explorer © Copyright IBM Corporation 2017

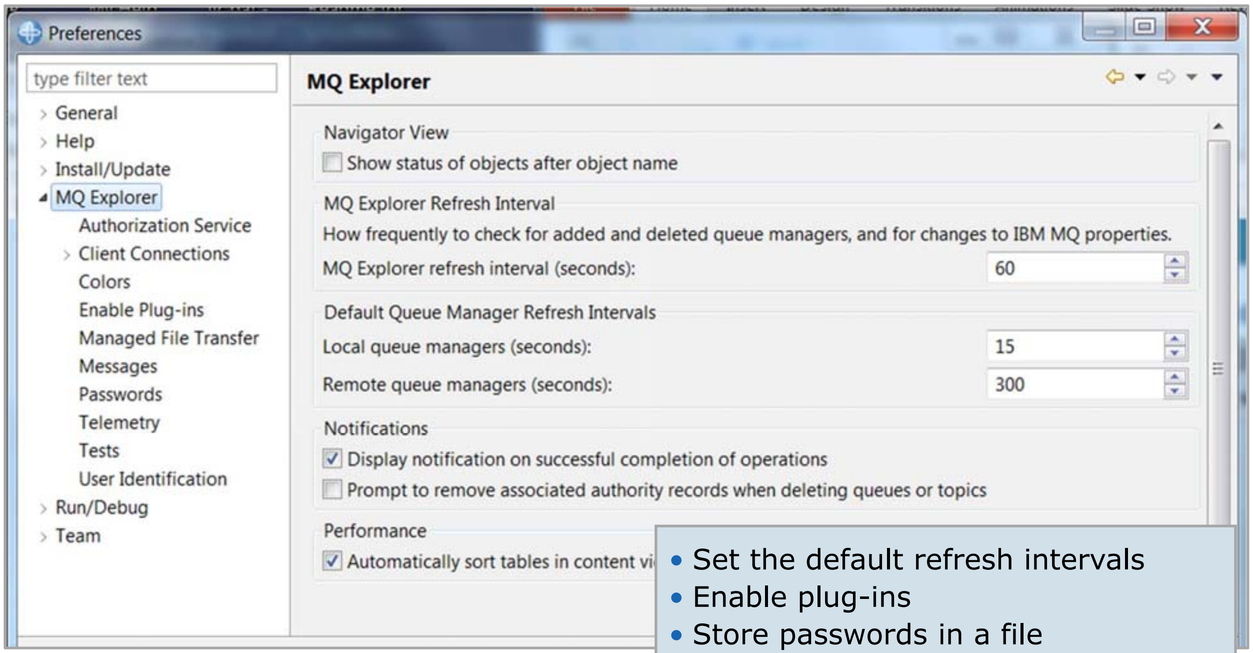
Figure 4-36. IBM MQ test results example

This figure is an example of the MQ test results.

An entry is provided for each test, which is identified by a **Category**. The test results are listed in order with all tests that generated errors listed first, followed by all tests that produced warnings, and finally, all tests that completed successfully.

IBM Training

IBM MQ Explorer preferences



Preferences

type filter text

- General
- Help
- Install/Update
- MQ Explorer
  - Authorization Service
  - Client Connections
  - Colors
  - Enable Plug-ins
  - Managed File Transfer
  - Messages
  - Passwords
  - Telemetry
  - Tests
  - User Identification
- Run/Debug
- Team

MQ Explorer

Navigator View

☐ Show status of objects after object name

MQ Explorer Refresh Interval

How frequently to check for added and deleted queue managers, and for changes to IBM MQ properties.

MQ Explorer refresh interval (seconds): 60

Default Queue Manager Refresh Intervals

Local queue managers (seconds): 15

Remote queue managers (seconds): 300

Notifications

✓ Display notification on successful completion of operations

☐ Prompt to remove associated authority records when deleting queues or topics

Performance

✓ Automatically sort tables in content vi

- Set the default refresh intervals
- Enable plug-ins
- Store passwords in a file
- Define default security exits
- Include SYSTEM objects during tests

Introduction to IBM MQ Explorer © Copyright IBM Corporation 2017

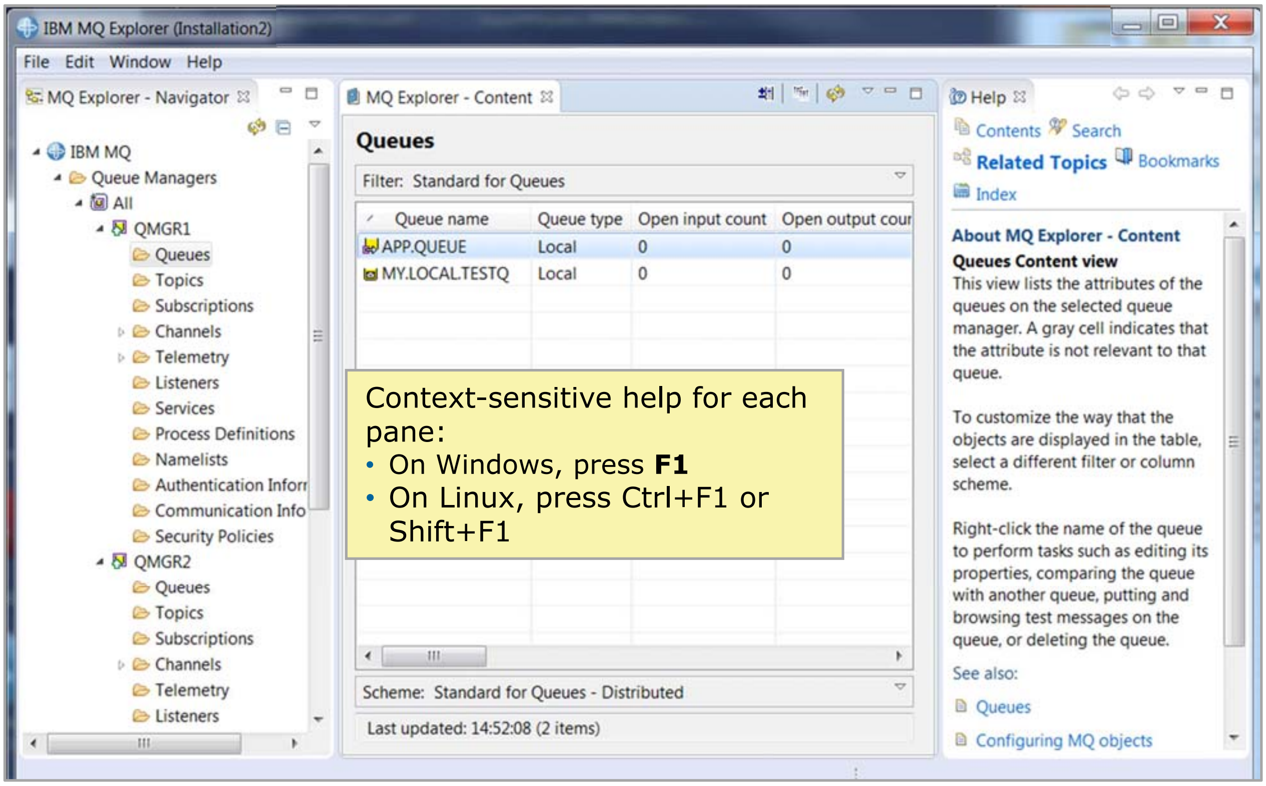
Figure 4-37. IBM MQ Explorer preferences

MQ Explorer preferences can be used to change some of the default behaviors and display options. Preferences include default refresh intervals and enabled plug-ins.

To display the MQ Explorer preferences, click **Windows > Preferences**.

IBM Training

Context-sensitive help



Queues

Filter: Standard for Queues

Queue name	Queue type	Open input count	Open output cour
APP.QUEUE	Local	0	0
MY.LOCAL.TESTQ	Local	0	0

Scheme: Standard for Queues - Distributed

Last updated: 14:52:08 (2 items)

Context-sensitive help for each pane:
- On Windows, press **F1**
- On Linux, press Ctrl+F1 or Shift+F1

About MQ Explorer - Content

Queues Content view

This view lists the attributes of the queues on the selected queue manager. A gray cell indicates that the attribute is not relevant to that queue.

To customize the way that the objects are displayed in the table, select a different filter or column scheme.

Right-click the name of the queue to perform tasks such as editing its properties, comparing the queue with another queue, putting and browsing test messages on the queue, or deleting the queue.

See also:

Queues

Configuring MQ objects

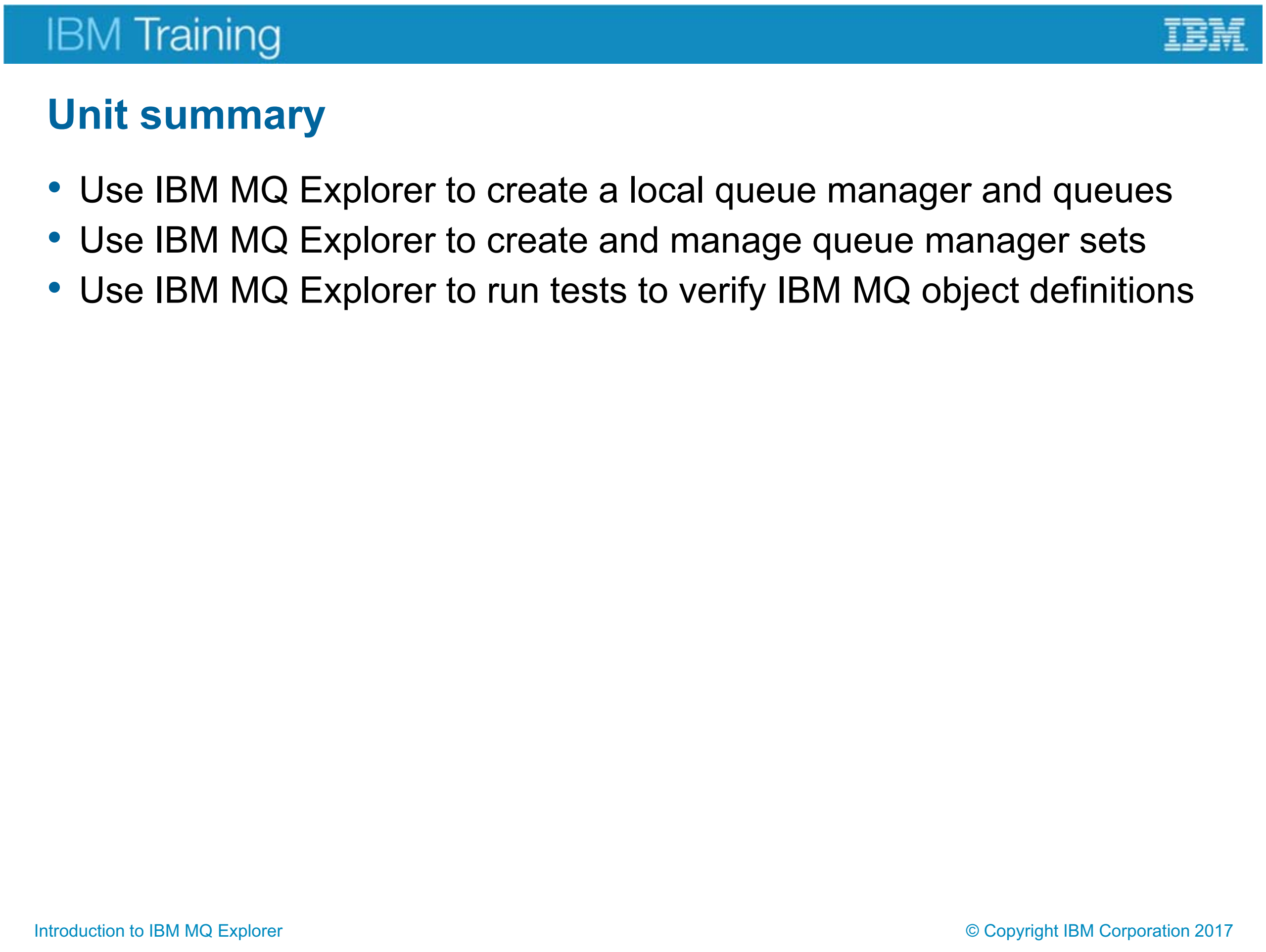
Introduction to IBM MQ Explorer © Copyright IBM Corporation 2017

Figure 4-38. Context-sensitive help

Quick access to reference documentation is supported in MQ Explorer by using context-sensitive help.

Context-sensitive help can be started by pressing **F1** on your keyboard while your cursor is over the required field.

The MQ Explorer **Help** preferences control the help information display mode. By default, the **Help** view is an extra view, but it can be changed to a window as shown in the figure, if you prefer. Help preferences are configured by clicking **Windows > Preferences > Help**.



IBM Training

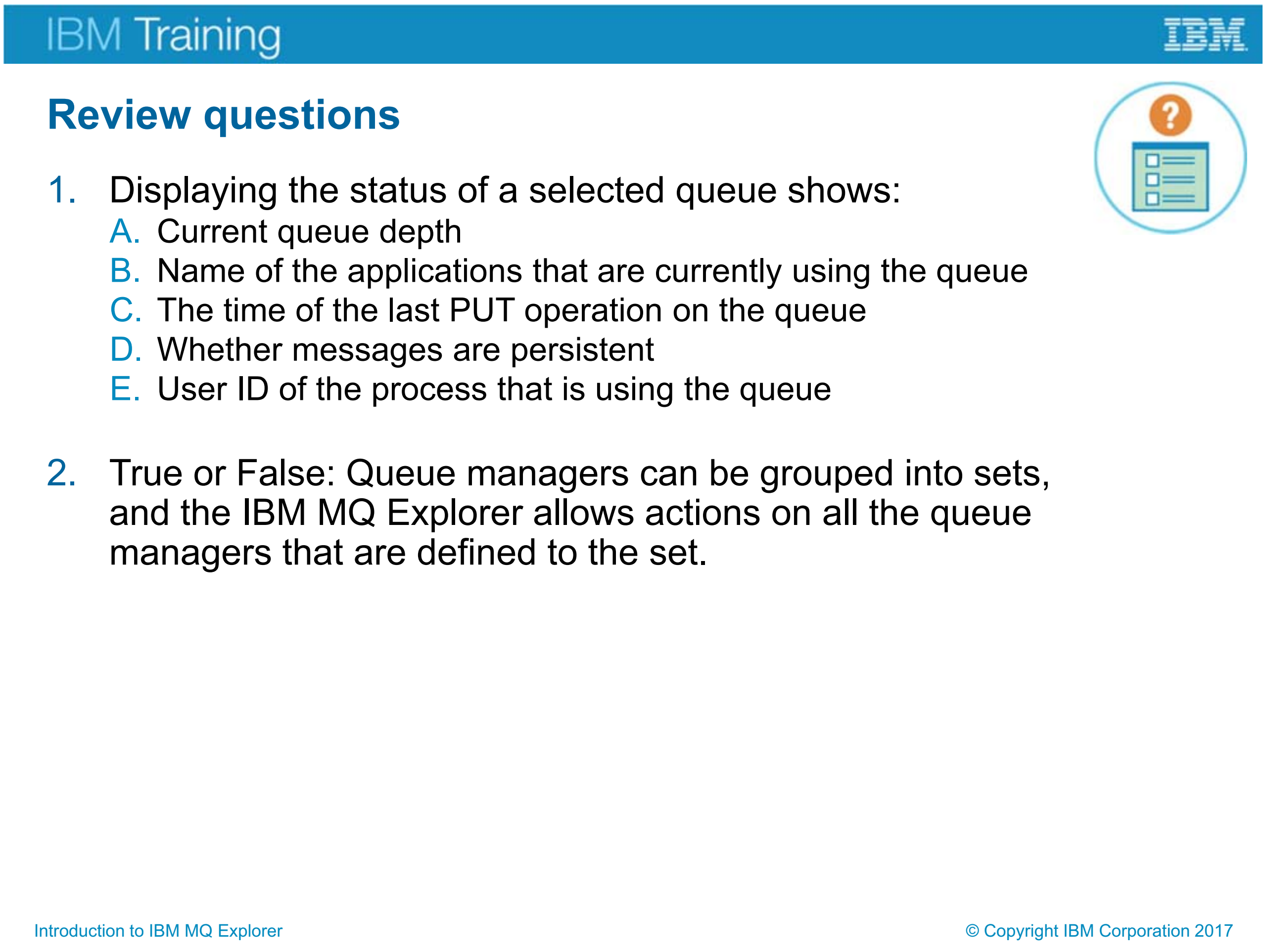
Unit summary

- Use IBM MQ Explorer to create a local queue manager and queues
- Use IBM MQ Explorer to create and manage queue manager sets
- Use IBM MQ Explorer to run tests to verify IBM MQ object definitions

Introduction to IBM MQ Explorer

© Copyright IBM Corporation 2017

Figure 4-39. Unit summary



IBM Training

Review questions

1. Displaying the status of a selected queue shows:
   A. Current queue depth
   B. Name of the applications that are currently using the queue
   C. The time of the last PUT operation on the queue
   D. Whether messages are persistent
   E. User ID of the process that is using the queue

2. True or False: Queue managers can be grouped into sets, and the IBM MQ Explorer allows actions on all the queue managers that are defined to the set.

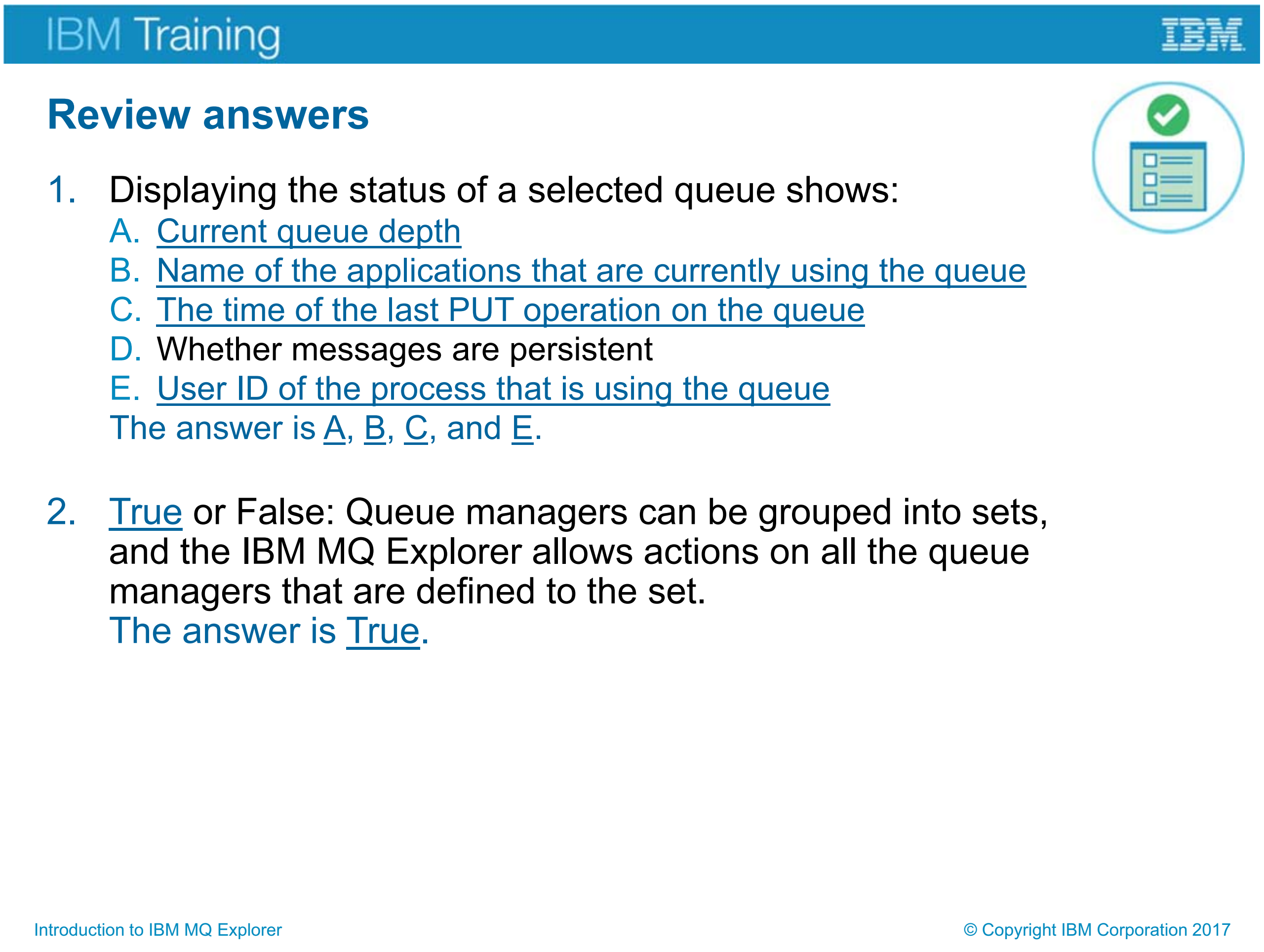
Introduction to IBM MQ Explorer

Figure 4-40. Review questions

Write your answers down here:

1.

2.



IBM Training

Review answers

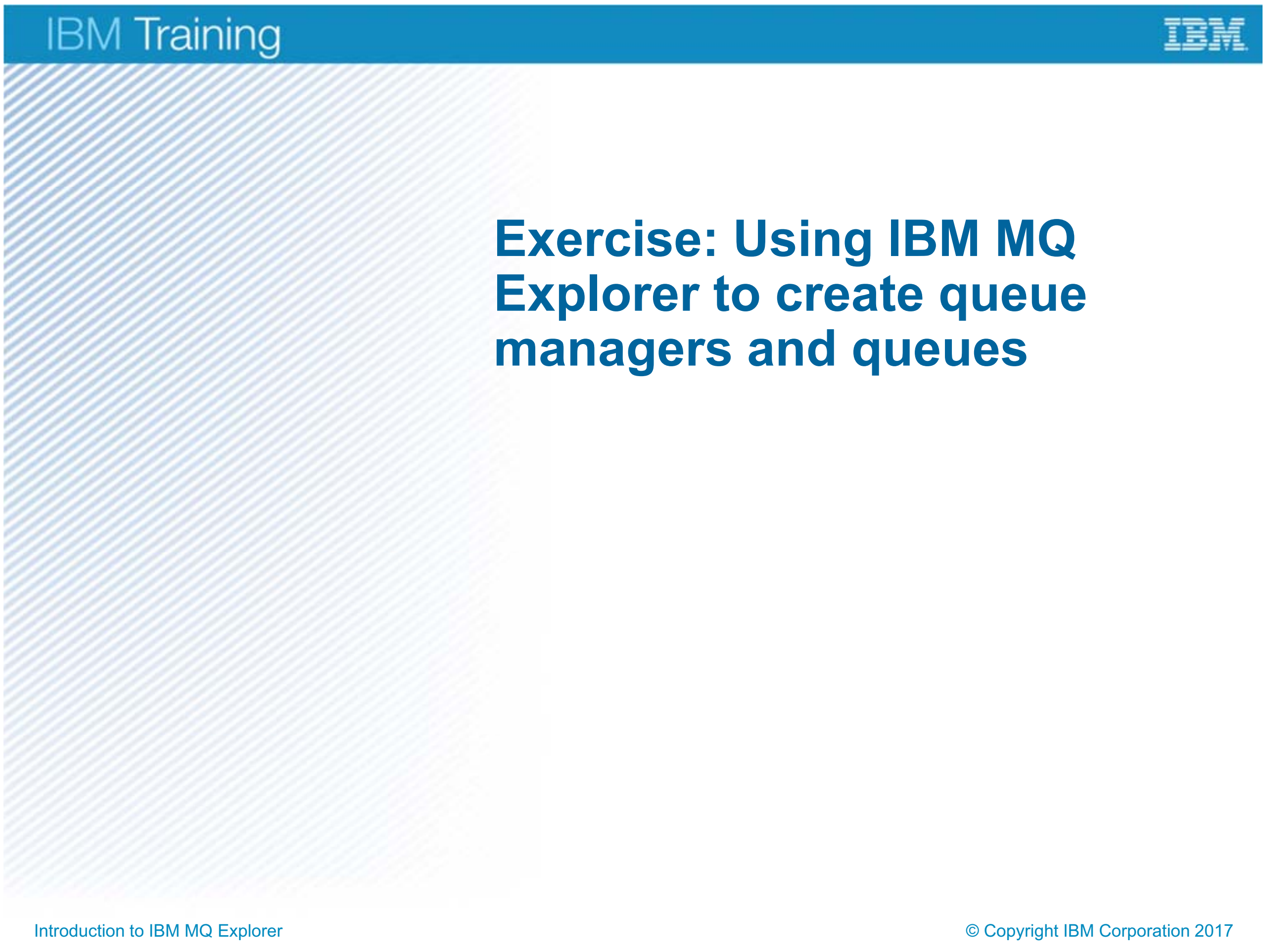
1. Displaying the status of a selected queue shows:
   A. Current queue depth
   B. Name of the applications that are currently using the queue
   C. The time of the last PUT operation on the queue
   D. Whether messages are persistent
   E. User ID of the process that is using the queue
   The answer is A, B, C, and E.

2. True or False: Queue managers can be grouped into sets, and the IBM MQ Explorer allows actions on all the queue managers that are defined to the set.
   The answer is True.

Introduction to IBM MQ Explorer

© Copyright IBM Corporation 2017

Figure 4-41. Review answers



IBM Training

IBM

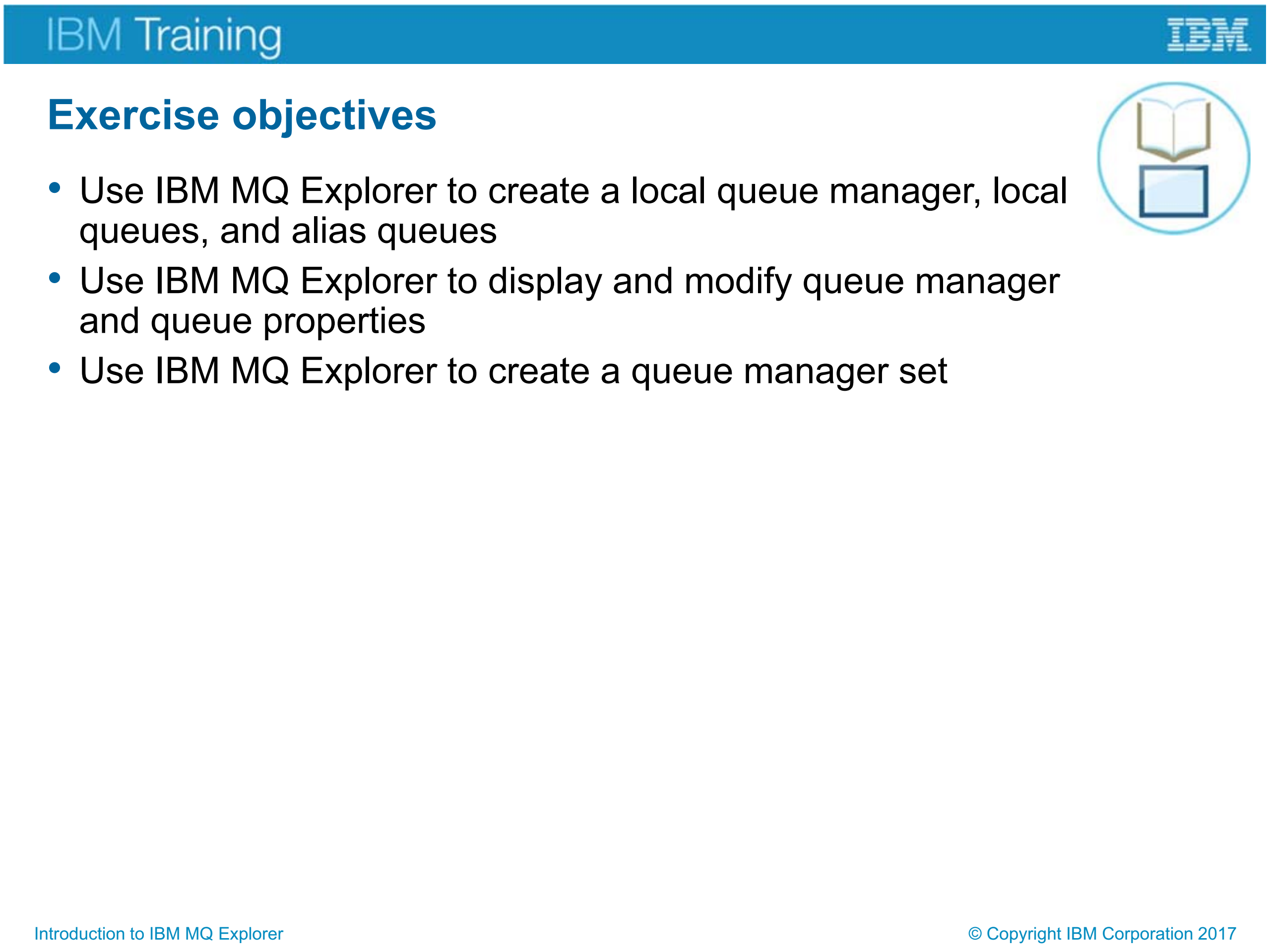
Exercise: Using IBM MQ Explorer to create queue managers and queues

Introduction to IBM MQ Explorer

© Copyright IBM Corporation 2017

Figure 4-42. Exercise: Using IBM MQ Explorer to create queue managers and queues

In this exercise, you use the IBM MQ Explorer to create a queue manager, start it, and then create queues. You also use IBM MQ Explorer to create queue manager sets to simplify the management of many queue managers.



IBM Training

Exercise objectives

- Use IBM MQ Explorer to create a local queue manager, local queues, and alias queues
- Use IBM MQ Explorer to display and modify queue manager and queue properties
- Use IBM MQ Explorer to create a queue manager set

Introduction to IBM MQ Explorer

© Copyright IBM Corporation 2017

Figure 4-43. Exercise objectives

See the *Course Exercises Guide* for the exercise description and detailed instructions.

Unit 5. Testing the IBM MQ implementation

Estimated time

00:30

Overview

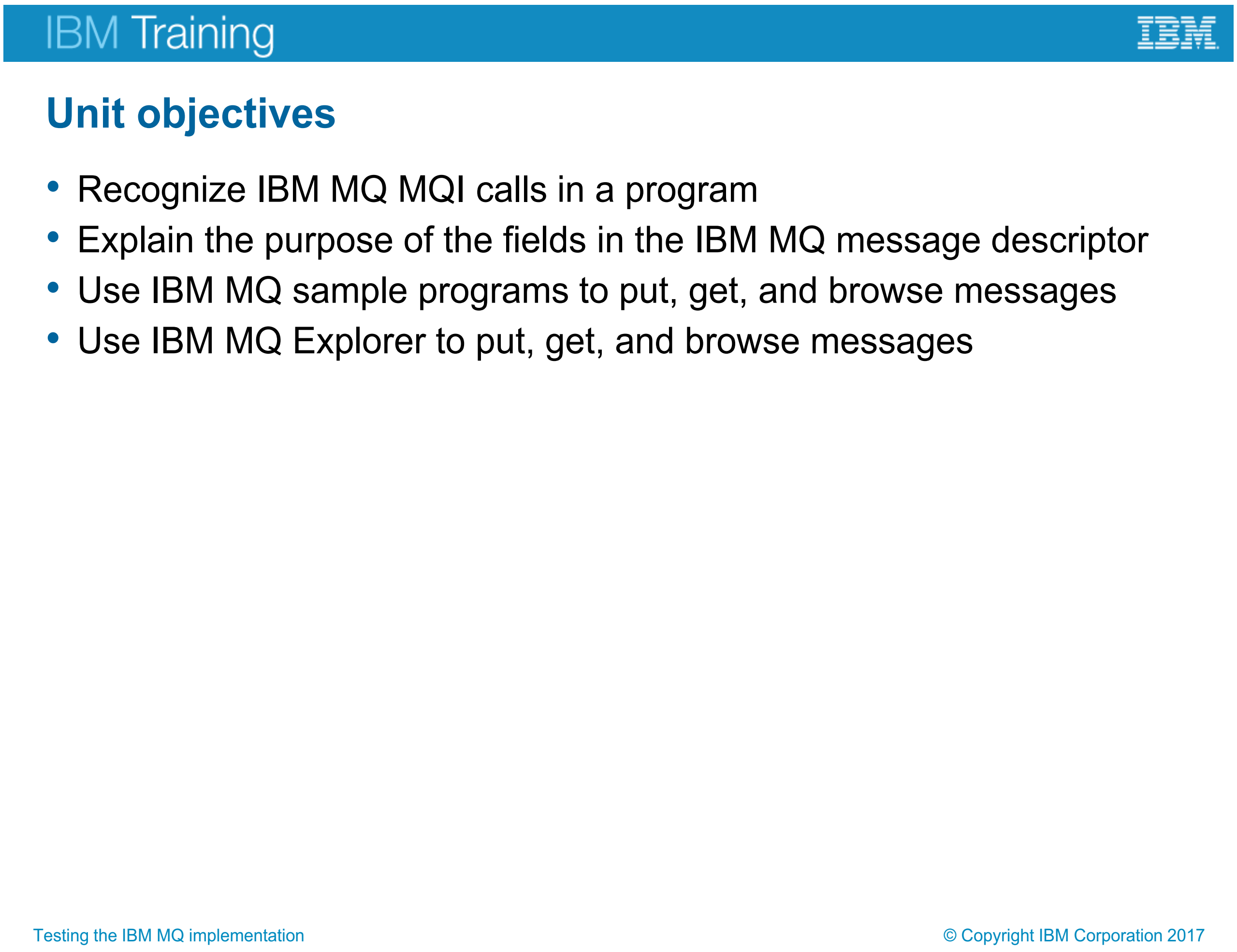
This unit provides a brief introduction to the IBM MQ Message Queue Interface (MQI). You also learn about the IBM MQ sample programs that you can use to test IBM MQ applications and the network.

How you will check your progress

- Checkpoint
- Hands-on exercises

References

IBM MQ product documentation



IBM Training

Unit objectives

- Recognize IBM MQ MQI calls in a program
- Explain the purpose of the fields in the IBM MQ message descriptor
- Use IBM MQ sample programs to put, get, and browse messages
- Use IBM MQ Explorer to put, get, and browse messages

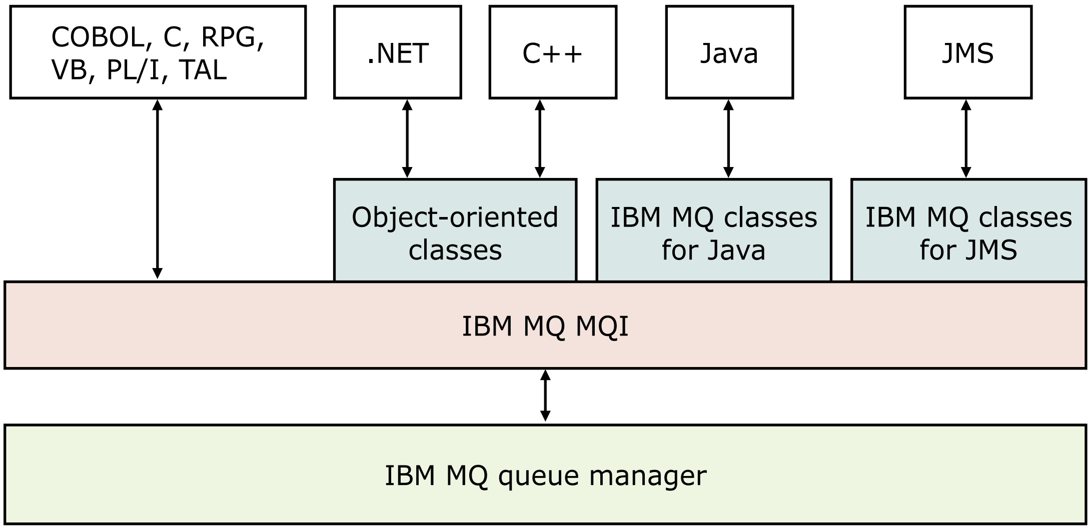
Testing the IBM MQ implementation

© Copyright IBM Corporation 2017

Figure 5-1. Unit objectives

IBM Training

IBM MQ Message Queue Interface (MQI)



- Procedural languages use the MQI to access message queuing services that IBM MQ queue managers provide

Figure 5-2. IBM MQ Message Queue Interface (MQI)

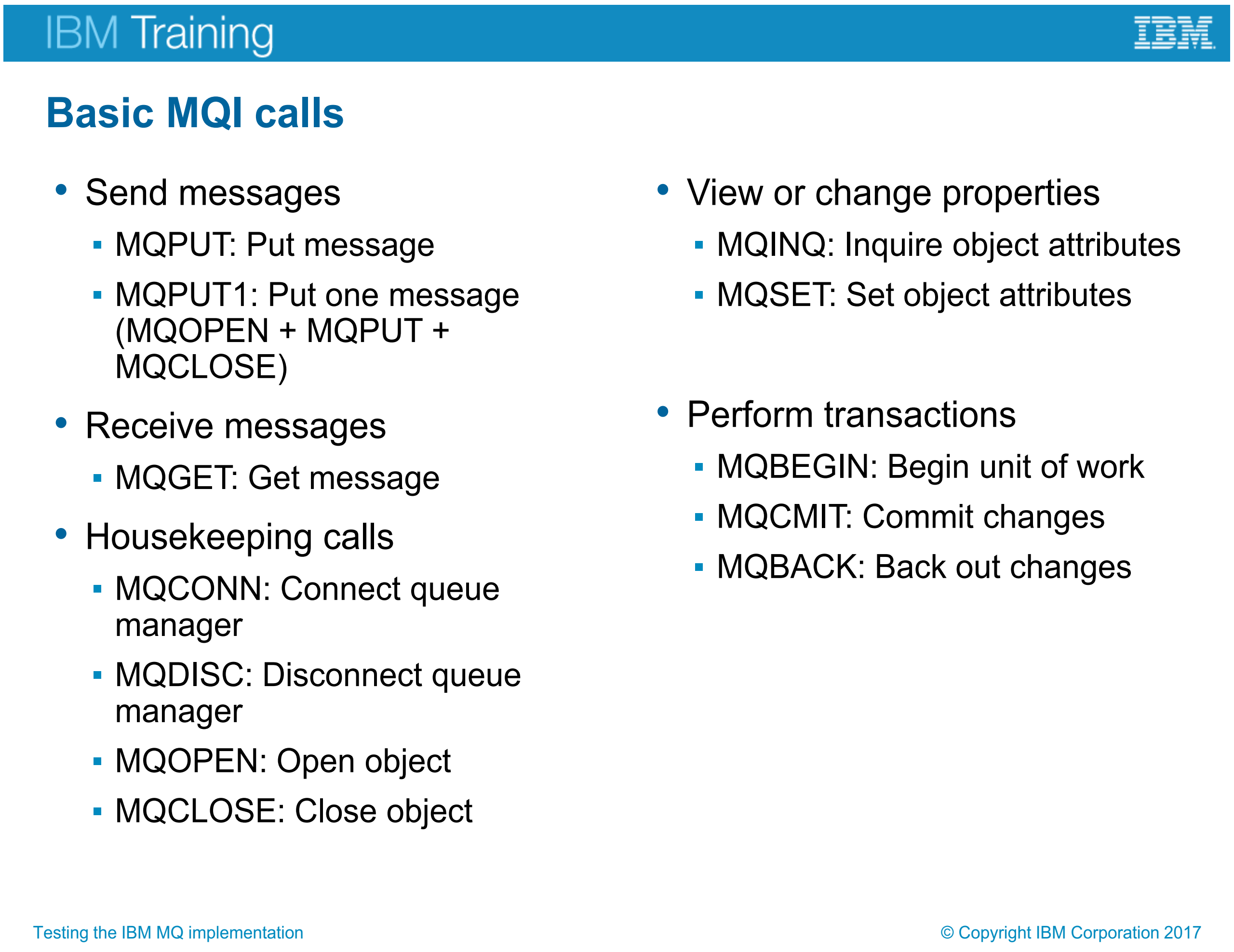
The IBM MQ Message Queue Interface (MQI) allows an application to access its queues and the messages they contain. The MQI is a simple application programming interface that is consistent across all operating systems that MQ supports. The MQI effectively protects applications from needing to know how a queue manager physically manages messages and queues. The MQI allows full access to MQ messaging support.

A common application programming interface across all supported operating systems is one of the major benefits of MQ.

You can develop client applications by using the MQ classes for .NET, MQ classes for Java, or MQ classes for Java Message Service (JMS).

MQ classes for .NET allow a program that is written in the .NET programming framework to connect to MQ as an MQ MQI client or to connect directly to an MQ server.

JMS is a specification of a portable application programming interface (API) for asynchronous messaging. JMS is an object-oriented Java API with a set of generic messaging objects for programmers to write event-based messaging applications. You can use Java and JMS clients on IBM i, UNIX, Linux, and Windows.



IBM Training

Basic MQI calls

- Send messages
  - MQPUT: Put message
  - MQPUT1: Put one message (MQOPEN + MQPUT + MQCLOSE)
- Receive messages
  - MQGET: Get message
- Housekeeping calls
  - MQCONN: Connect queue manager
  - MQDISC: Disconnect queue manager
  - MQOPEN: Open object
  - MQCLOSE: Close object
- View or change properties
  - MQINQ: Inquire object attributes
  - MQSET: Set object attributes
- Perform transactions
  - MQBEGIN: Begin unit of work
  - MQCMIT: Commit changes
  - MQBACK: Back out changes

Testing the IBM MQ implementation

© Copyright IBM Corporation 2017

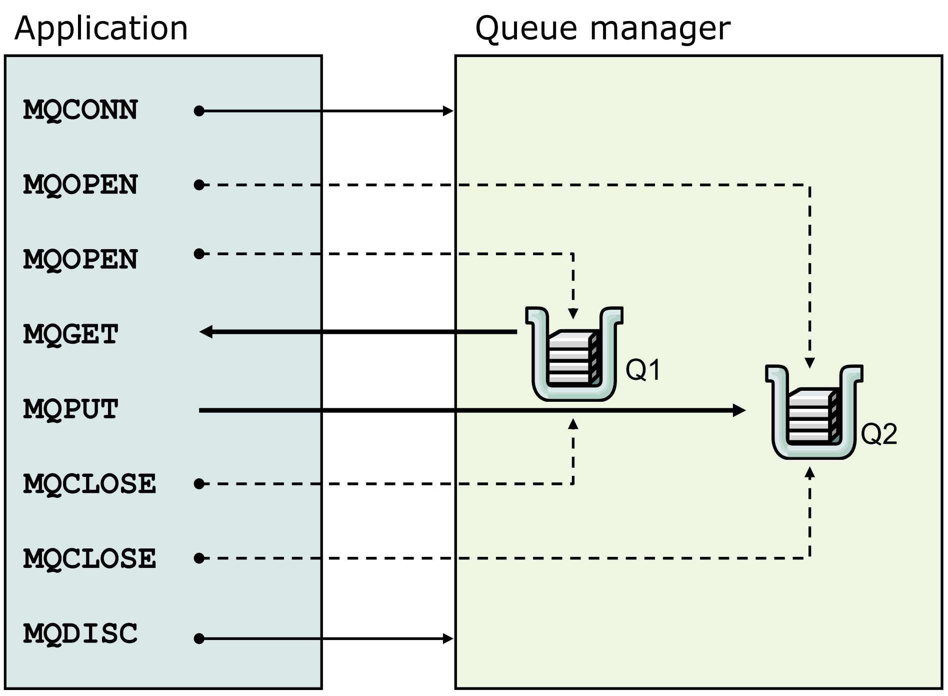
Figure 5-3. Basic MQI calls

By using the most basic MQI calls, an application can put a message on a queue and get a message from a queue. The MQI calls are described in detail in the production documentation.

- **MQPUT** and **MQPUT1** put a message on a named queue. Generally, a message is added to the end of a queue.
- **MQGET** gets a message from a named queue. Generally, a message is removed from the front of a queue.
- **MQCONN**, **MQCONNX**, and **MQDISC** are used by an application to connect to and disconnect from a queue manager. An application must connect to a queue manager before it can send any more MQI calls.
- **MQOPEN** and **MQCLOSE** open a queue for specified operations and close the queue when access to it is no longer required. An application must open a queue before it can put messages on it, or get messages from it.
- **MQINQ** and **MQSET** inquire and set the attributes of an object. All MQ objects, such as a queue, a process, and the queue manager object, have a set of attributes.
- **MQBEGIN**, **MQCMIT**, and **MQBACK** allow an application to put and get messages as part of a unit of work.

IBM Training

Basic MQI calls in use



Testing the IBM MQ implementation

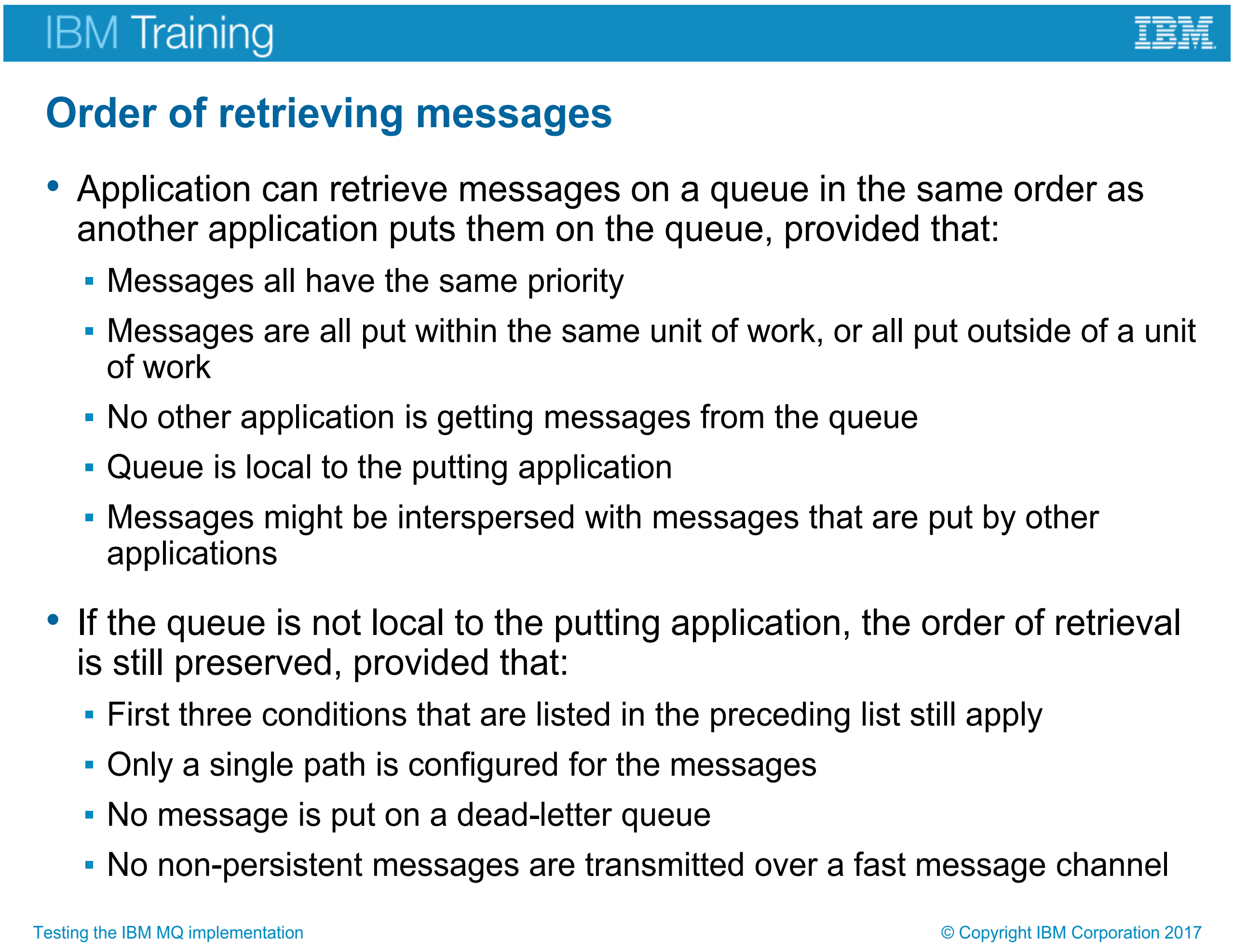
Figure 5-4. Basic MQI calls in use

The MQI is a simple call interface. The figure shows the most common calls that are used to connect and disconnect to a queue manager, and to put and get messages from a queue.

In a typical sequence:

- MQCONN connects to the queue manager.
- MQOPEN establishes access to an MQ object such as a queue.
- MQGET retrieves a message from a queue.
- MQPUT places a message on a queue or publishes to a new topic.
- MQCLOSE releases access to an MQ object.
- MQDISC disconnects from the queue manager.

MQ application development is taught in course WM513, *IBM MQ V9 Application Programming*.



IBM Training

Order of retrieving messages

- Application can retrieve messages on a queue in the same order as another application puts them on the queue, provided that:
  - Messages all have the same priority
  - Messages are all put within the same unit of work, or all put outside of a unit of work
  - No other application is getting messages from the queue
  - Queue is local to the putting application
  - Messages might be interspersed with messages that are put by other applications
- If the queue is not local to the putting application, the order of retrieval is still preserved, provided that:
  - First three conditions that are listed in the preceding list still apply
  - Only a single path is configured for the messages
  - No message is put on a dead-letter queue
  - No non-persistent messages are transmitted over a fast message channel

Testing the IBM MQ implementation © Copyright IBM Corporation 2017

Figure 5-5. Order of retrieving messages

Most applications do not have a strict need to process messages in the same order as they were created, but some applications might have such a requirement, even a legal obligation to do so.

MQ documents the conditions under which an application can retrieve a sequence of messages from a queue in the same order that another application puts them. The conditions assume that the application that retrieves the sequence of messages is not using selection criteria. If the conditions are not met, applications must either include sequencing information in the application data of a message or establish a means of acknowledging receipt of a message before the next one is sent.

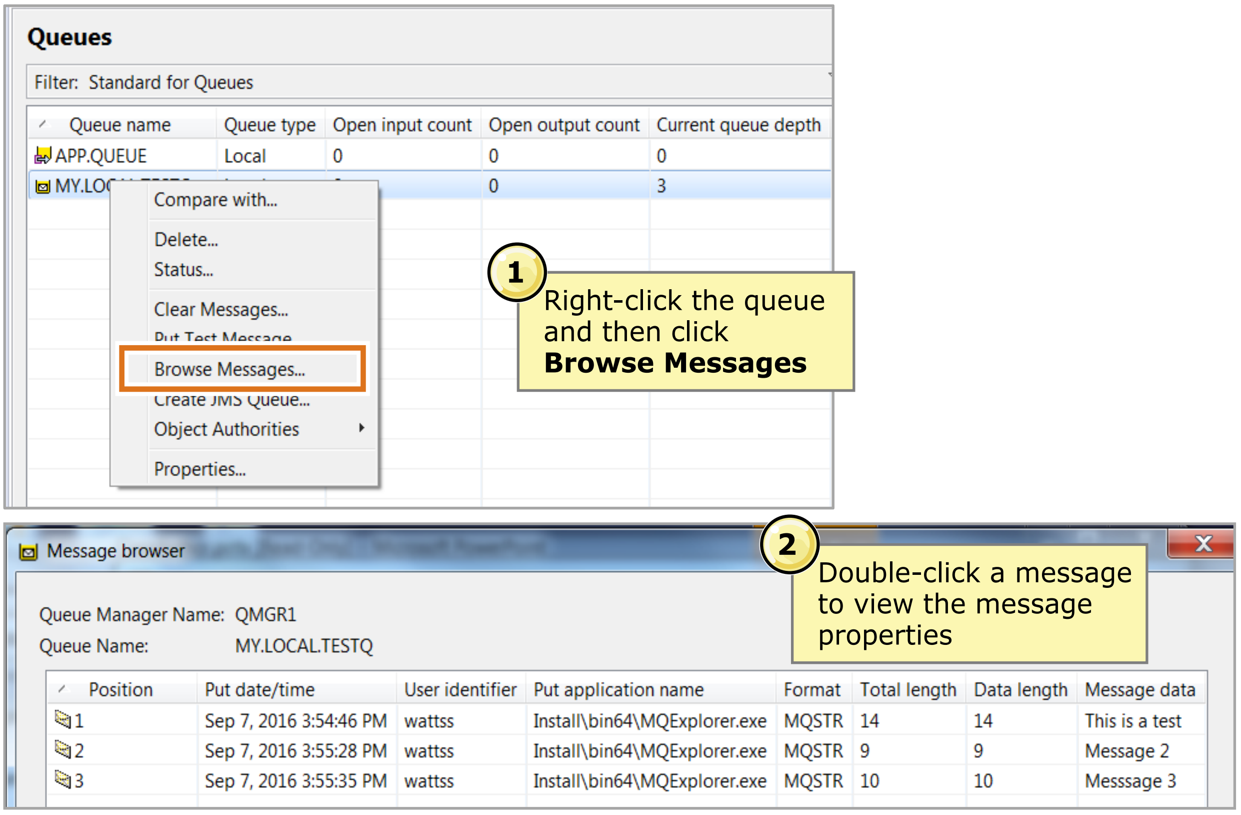
The first set of conditions in the figure assumes that the application that puts the messages and the application that gets the messages are both connected to the same queue manager, so no remote queuing is involved.

The second set of conditions describes the process when the “putting” and “getting” applications are connected to different queue managers.

On a fast message channel, nonpersistent messages overtake persistent messages on the same channel and so arrive out of sequence. This behavior is because the receiving MCA commits a nonpersistent message on its destination queue as soon as it receives it instead of waiting for the end of the batch.

IBM Training

Browsing messages in IBM MQ Explorer



Queue name	Queue type	Open input count	Open output count	Current queue depth
APP.QUEUE	Local	0	0	0
MY.LOC			0	3

Queue Manager Name: QMGR1
Queue Name: MY.LOCAL.TESTQ

Position	Put date/time	User identifier	Put application name	Format	Total length	Data length	Message data
1	Sep 7, 2016 3:54:46 PM	wattss	Install\bin64\MQExplorer.exe	MQSTR	14	14	This is a test
2	Sep 7, 2016 3:55:28 PM	wattss	Install\bin64\MQExplorer.exe	MQSTR	9	9	Message 2
3	Sep 7, 2016 3:55:35 PM	wattss	Install\bin64\MQExplorer.exe	MQSTR	10	10	Messsage 3

Testing the IBM MQ implementation © Copyright IBM Corporation 2017

Figure 5-6. Browsing messages in IBM MQ Explorer

You can browse the queue at any time to see the messages that are on a queue.

This figure and the next show the steps for browsing the contents of a queue in MQ Explorer.

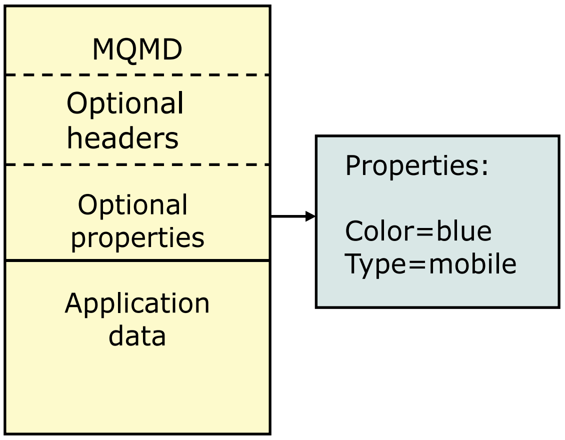
1. Click **Queues** in the **MQ Explorer - Navigator** view to display the **Queues** content view.
2. Right-click the queue and then click **Browse Messages**.
3. The **Message browser** view provides a summary of each message that is on the queue.

   For more information about a specific message, double-click the message in the **Message browser** view.

IBM Training

Message properties

- Attributes of IBM MQ messages that are associated with the message but not part of the body
  - Consist of a textual name and a value of a particular type
- Properties can be integers, strings, Booleans, and more
- Allows explicit statement of relationships between messages
- Can be viewed in IBM MQ Explorer



Testing the IBM MQ implementation

Figure 5-7. Message properties

Message selectors use message properties to filter publications to topics or to selectively get messages from queues. Message properties are optional. They can be used to include business data or state information without storing it in the message data.

Message properties are arbitrary name-value pairs that can be assigned to a message. Properties are assigned by using two MQI verbs: MQSETMP to set the properties and MQINQMP to inquire on the message property value.

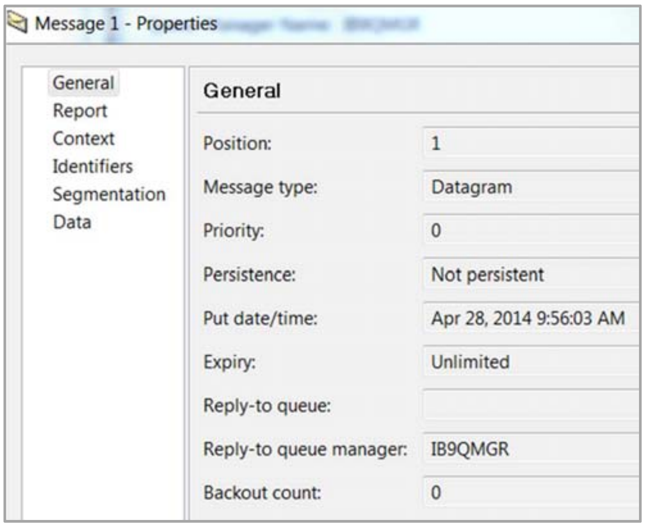
Message properties are primarily used for establishing explicit relationships between messages by supplementing the message ID and correlation ID. For example, message properties can specify that *message X* is a reply to *message Y*.

You can view the message properties in MQ Explorer or by using the WebSphere MQ SupportPac RFHUtil.

IBM Training

General message properties

- **Position**: Position in queue
- **Message type**: Type of message
- **Priority**: Priority of message
- **Persistence**: Indicates whether message is persistent or not persistent
- **Put date/time**: Date and time message was put on queue
- **Expiry**: Time after which message is eligible to be discarded
- **Reply-to queue**: Name of message queue to which application that gets the message should send reply messages
- **Reply-to queue manager**: Name of queue manager on which reply-to queue is defined
- **Backout count**: Number of times MQGET call returned the message as part of a unit of work, and then backed out

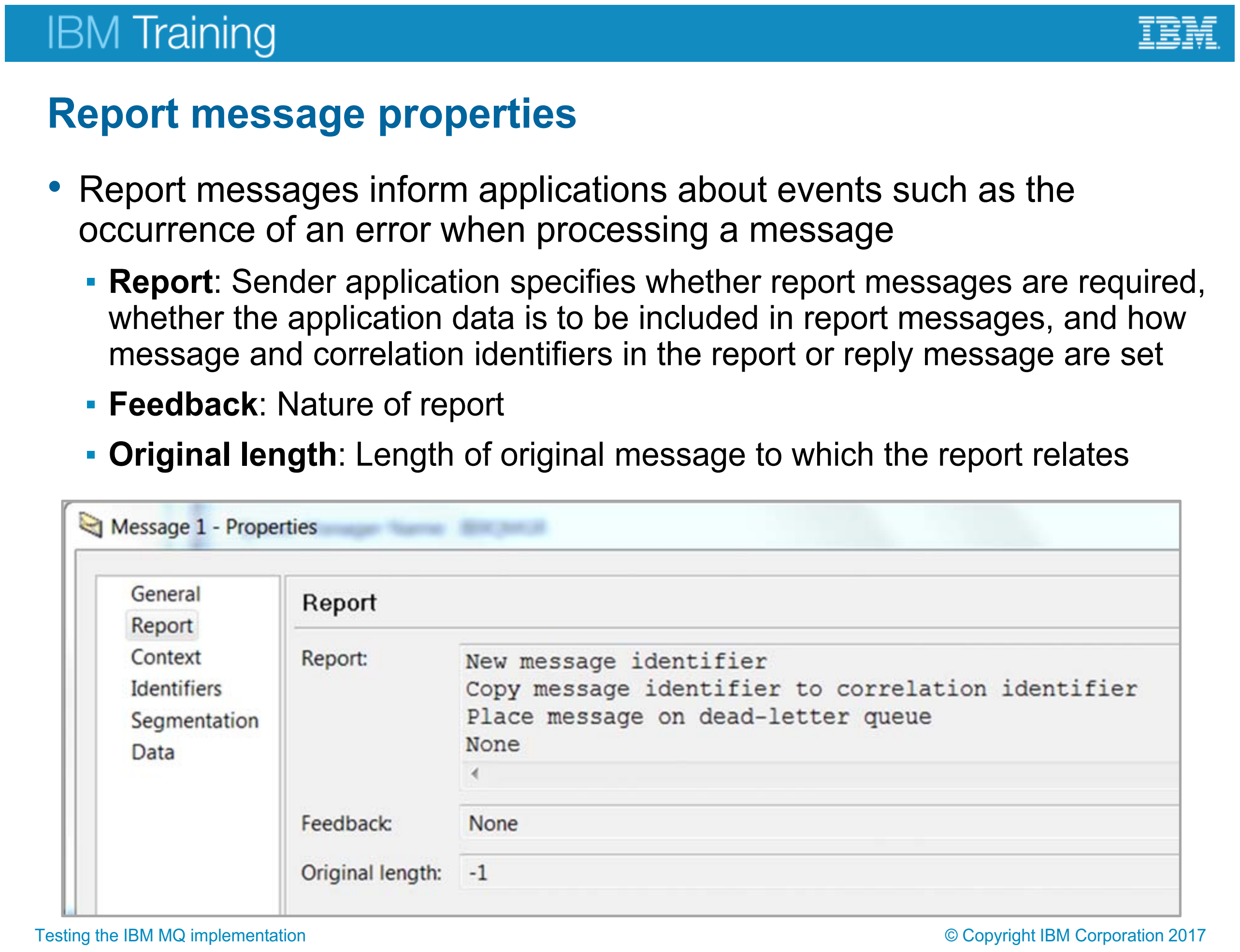


Position:	1
Message type:	Datagram
Priority:	0
Persistence:	Not persistent
Put date/time:	Apr 28, 2014 9:56:03 AM
Expiry:	Unlimited
Reply-to queue:	
Reply-to queue manager:	IB9QMGR
Backout count:	0

Figure 5-8. General message properties

You can view parts of the MQMD and other properties of the message in MQ Explorer. This figure provides an overview of the **General** properties in the MQ Explorer message browser.

The application sets some properties, such as message type, priority, and persistence, in the MQMD.



IBM Training

Report message properties

- Report messages inform applications about events such as the occurrence of an error when processing a message
  - **Report**: Sender application specifies whether report messages are required, whether the application data is to be included in report messages, and how message and correlation identifiers in the report or reply message are set
  - **Feedback**: Nature of report
  - **Original length**: Length of original message to which the report relates

Message 1 - Properties

General
Report
Context
Identifiers
Segmentation
Data

Report

Report:
```
New message identifier
Copy message identifier to correlation identifier
Place message on dead-letter queue
None
```

Feedback: None

Original length: -1

Testing the IBM MQ implementation © Copyright IBM Corporation 2017

Figure 5-9. Report message properties

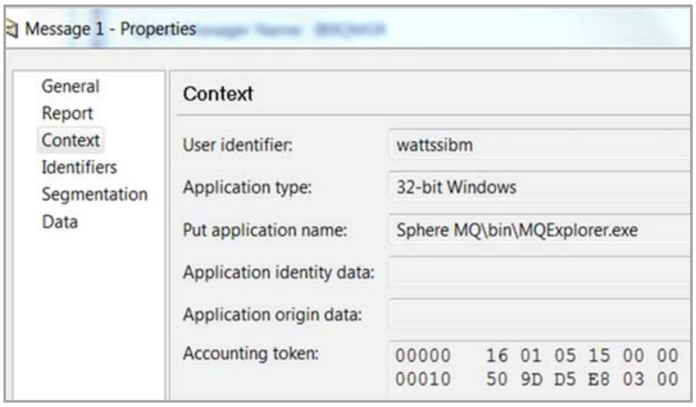
A report message is a message about another message. It is used to inform the application about expected or unexpected events that relate to the original message.

The **Report** properties page displays the attributes that are related to report messages. This figure provides an overview of the **Report** properties in the MQ Explorer message browser.

IBM Training

Context message properties

- Context information allows application that retrieves message to determine originator
- *Identity context* identifies user of the application that first put the message
  - **User identifier** of application that originated message
  - **Accounting token** allows application to charge for work
  - **Application identity data** that application defines to provide information about the message or its originator
- *Origin context* describes the application that put the message on the current queue
  - **Application type** that put the message
  - **Put application name** of the application that put the message
  - **Application origin data** that application defines to provide more information about message origin


Message 1 - Properties

General
Report
Context
Identifiers
Segmentation
Data

Context

User identifier:	wattssibm
Application type:	32-bit Windows
Put application name:	Sphere MQ\bin\MQExplorer.exe
Application identity data:	
Application origin data:	
Accounting token:	00000 16 01 05 15 00 00 00010 50 9D D5 E8 03 00

Testing the IBM MQ implementation © Copyright IBM Corporation 2017

Figure 5-10. Context message properties

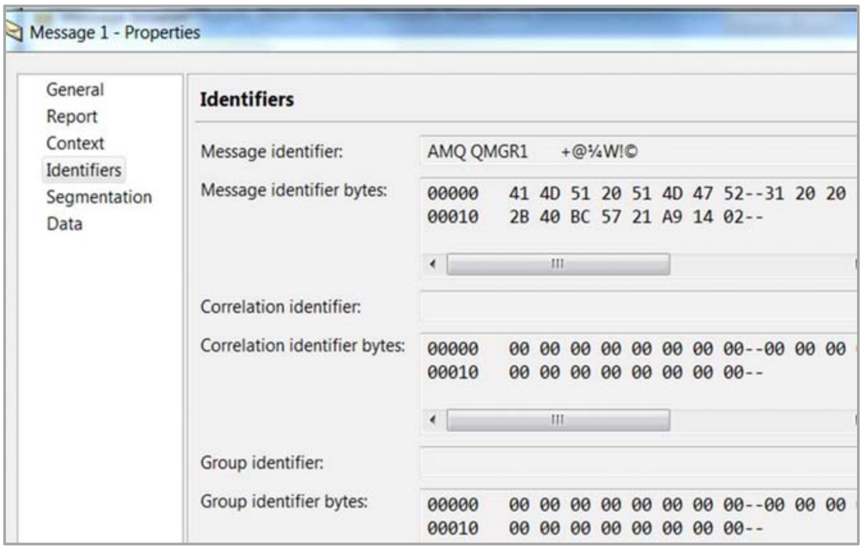
The **Context** message properties page displays information from the sender application about the message.

This figure provides an overview of the message properties that are contained on the **Context** message properties page.

IBM Training

Identifiers message properties

- Displays identification information that is associated with the message
  - **Message identifier**: Distinguishes one message from another
  - **Message identifier bytes**: Message identifier in byte form
  - **Correlation identifier**: Identifier that application can use to relate one message to another, or to relate message to other work that the application completes
  - **Correlation identifier bytes**: Correlation identifier in byte form
  - **Group identifier**: Identifies particular message group or logical message to which the physical message belongs
  - **Group identifier bytes**: Group identifier in byte form



Message identifier: AMQ QMGR1 +@¼W!©

Message identifier bytes:
```
00000    41 4D 51 20 51 4D 47 52--31 20 20
00010    2B 40 BC 57 21 A9 14 02--
```

Correlation identifier:

Correlation identifier bytes:
```
00000    00 00 00 00 00 00 00 00--00 00 00
00010    00 00 00 00 00 00 00 00--
```

Group identifier:

Group identifier bytes:
```
00000    00 00 00 00 00 00 00 00--00 00 00
00010    00 00 00 00 00 00 00 00--
```

Testing the IBM MQ implementation

© Copyright IBM Corporation 2017

Figure 5-11. Identifiers message properties

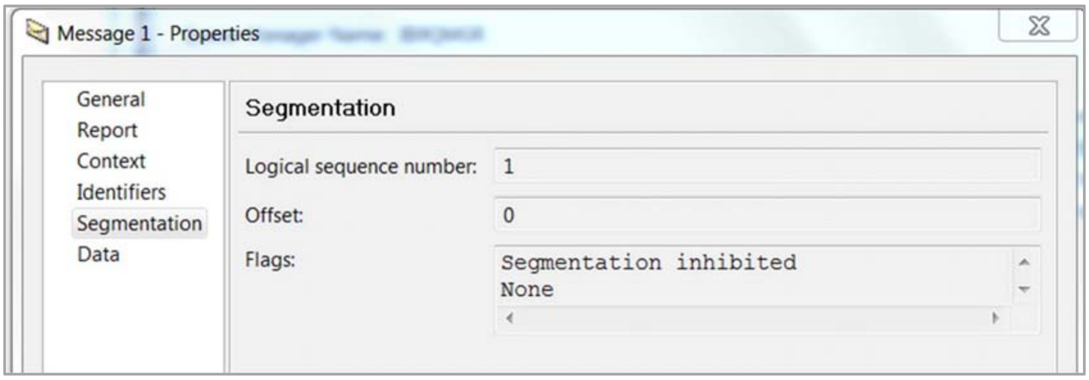
The **Identifiers** message properties page displays identification information that is associated with the message.

This figure provides an overview of the message properties that are contained on the **Identifiers** page in MQ Explorer.

IBM Training

Segmentation message properties

- Displays attributes related to segmenting large messages
  - **Logical sequence number**: Sequence number of logical message within the group
  - **Offset**: Offset of data in physical message from the start of the logical message
  - **Flags**: Message flags that specify attributes of message, or control its processing



Logical sequence number: 1

Offset: 0

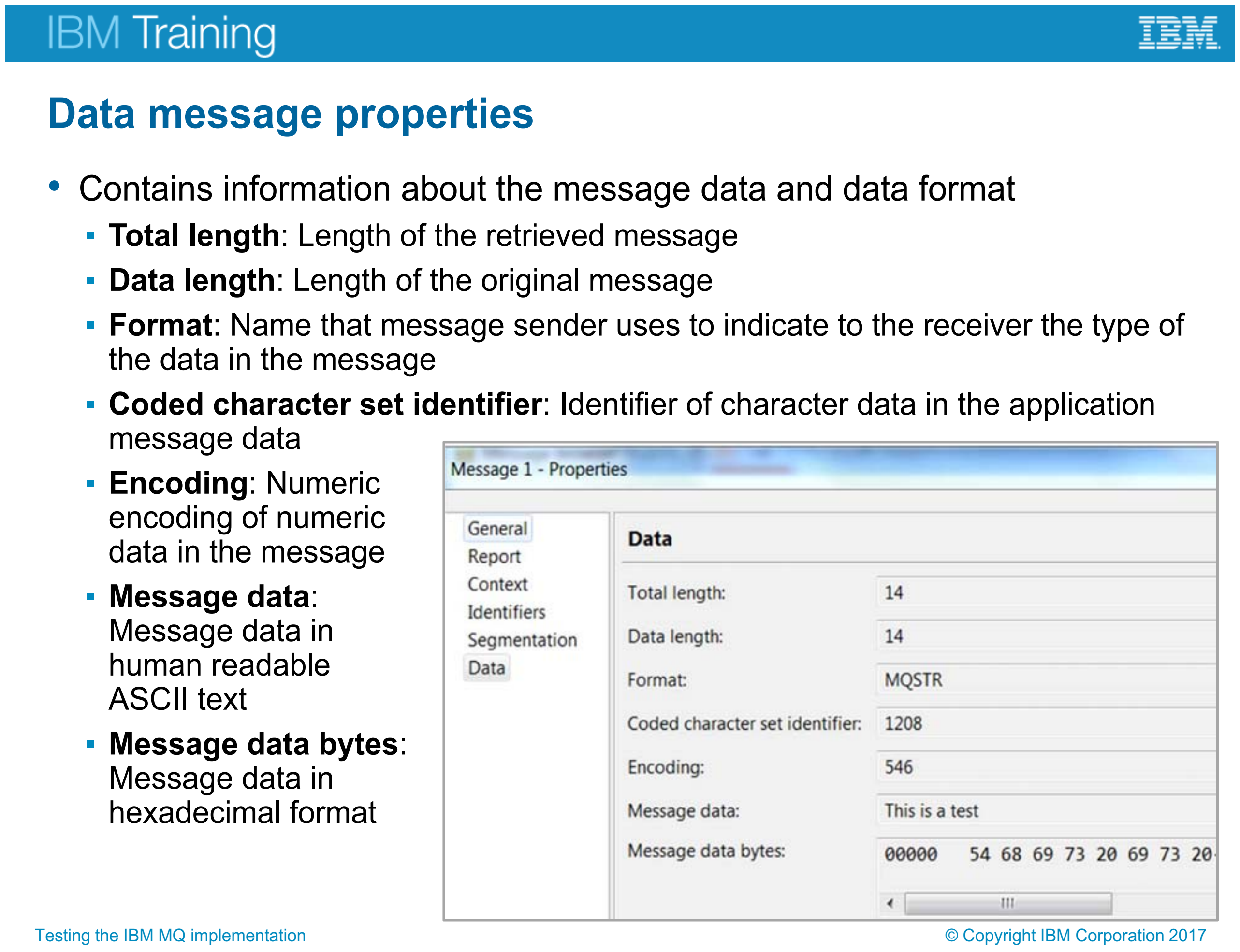
Flags: Segmentation inhibited None

Testing the IBM MQ implementation © Copyright IBM Corporation 2017

Figure 5-12. Segmentation message properties

The **Segmentation** message properties page in MQ Explorer displays the attributes that are related to segmenting large messages.

This figure provides an overview of the message properties that are contained on the **Segmentation** message properties page.



IBM Training

Data message properties

- Contains information about the message data and data format
  - **Total length**: Length of the retrieved message
  - **Data length**: Length of the original message
  - **Format**: Name that message sender uses to indicate to the receiver the type of the data in the message
  - **Coded character set identifier**: Identifier of character data in the application message data
  - **Encoding**: Numeric encoding of numeric data in the message
  - **Message data**: Message data in human readable ASCII text
  - **Message data bytes**: Message data in hexadecimal format

Message 1 - Properties

Data

Total length:	14
Data length:	14
Format:	MQSTR
Coded character set identifier:	1208
Encoding:	546
Message data:	This is a test
Message data bytes:	00000 54 68 69 73 20 69 73 20

Testing the IBM MQ implementation

© Copyright IBM Corporation 2017

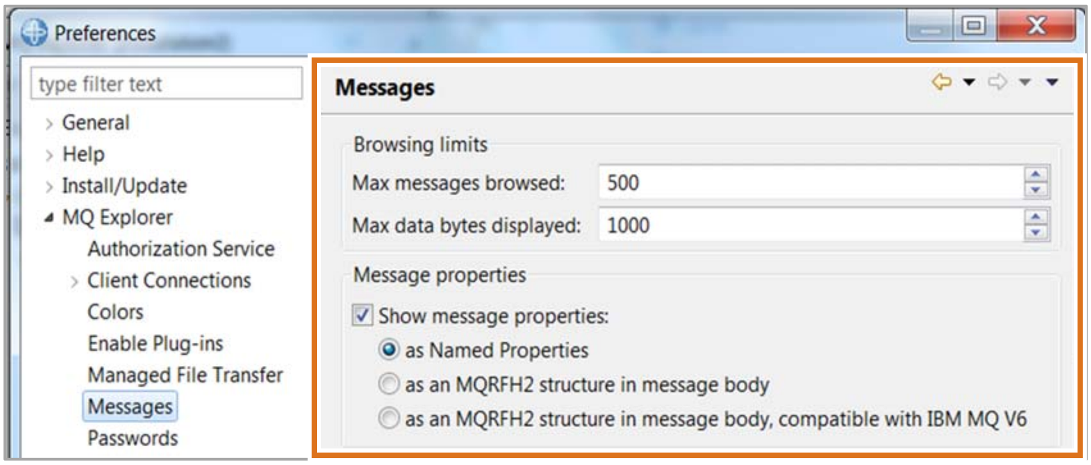
Figure 5-13. Data message properties

The **Data** message properties page in MQ Explorer displays the message data itself and information about the data format.

This figure provides an overview of the message properties on the **Data** message properties page.

IBM Training

Message property preferences in IBM MQ Explorer



Messages

Browsing limits

Max messages browsed: 500

Max data bytes displayed: 1000

Message properties

[✓] Show message properties:

[●] as Named Properties

[ ] as an MQRFH2 structure in message body

[ ] as an MQRFH2 structure in message body, compatible with IBM MQ V6

- Browsing limits
  - Maximum number of messages that can be browsed (1 – 5000)
  - Maximum number of bytes of data to display per message (0 – 16384)
- Show message properties not contained in MQMD
  - As name-value pairs
  - As an MQRFH2 header structure

Figure 5-14. Message property preferences in IBM MQ Explorer

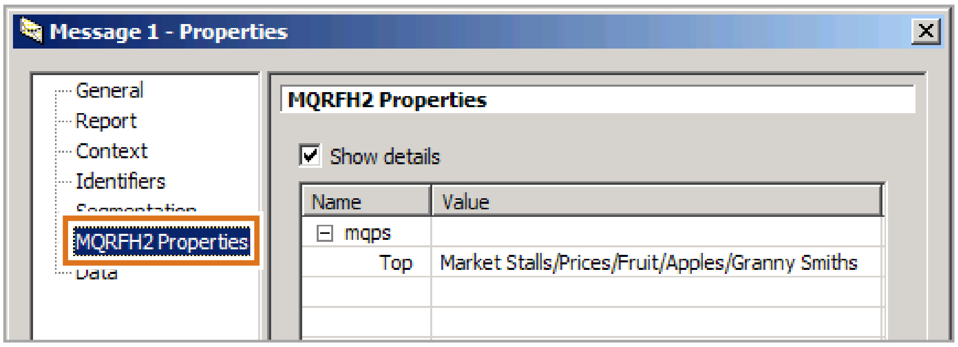
MQ Explorer can be configured to read properties in messages on the queue based on the message properties.

The way message properties are shown in MQ depends on the configuration of MQ Explorer and the definition of the queue.

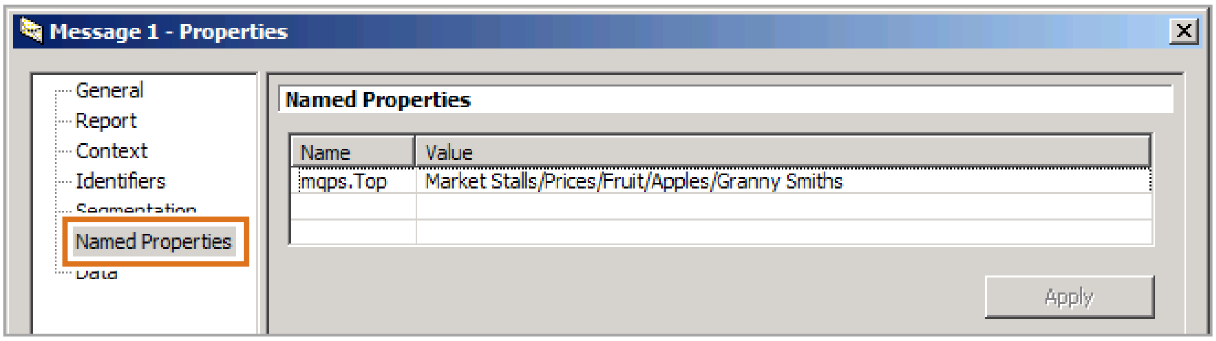
Depending on the **Preference** property setting, you see properties either as **Named Properties** or **MQRFH2 Properties** when you look at the properties of a browsed message.

IBM Training

Message property preferences: Examples



Name	Value
⊟ mqps	
Top	Market Stalls/Prices/Fruit/Apples/Granny Smiths



Name	Value
mqps.Top	Market Stalls/Prices/Fruit/Apples/Granny Smiths

Testing the IBM MQ implementation © Copyright IBM Corporation 2017

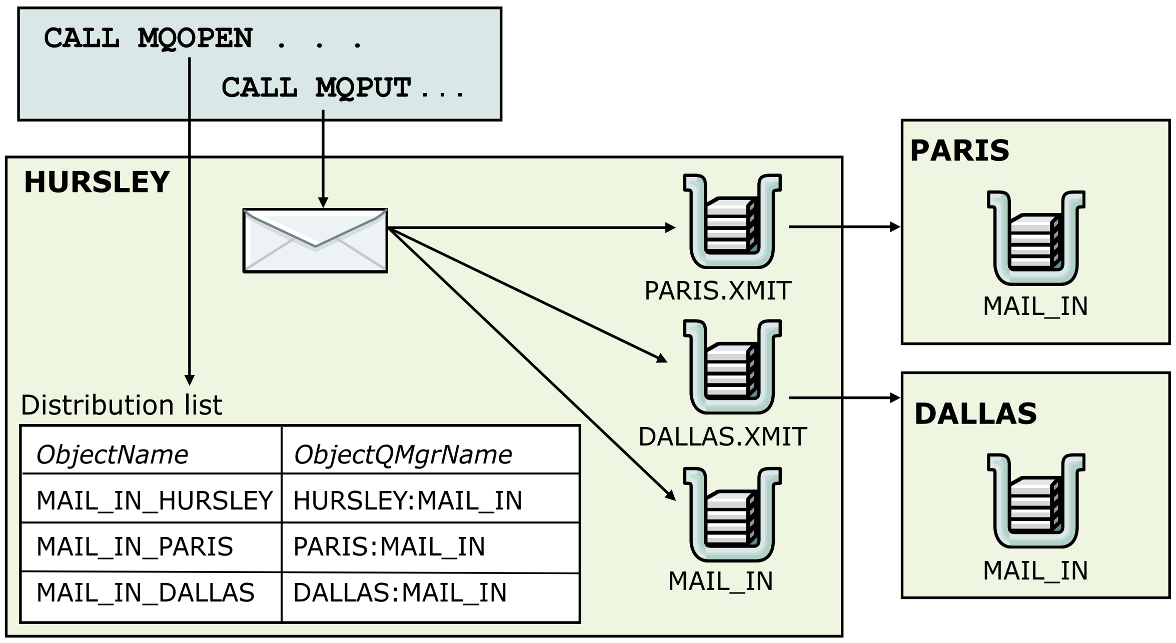
Figure 5-15. Message property preferences: Examples

This figure shows the two different ways that you can display message properties, based on your selection in the MQ Explorer **Messages Preferences**.

If you select the option to show message properties as named properties, the **Named Properties** message properties contain the name of the message property and the value of the named property.

IBM Training

Distribution lists



ObjectName	*ObjectQMgrName*
MAIL_IN_HURSLEY	HURSLEY:MAIL_IN
MAIL_IN_PARIS	PARIS:MAIL_IN
MAIL_IN_DALLAS	DALLAS:MAIL_IN

- Allow an application to send one MQOPEN call to open multiple queues and a one MQPUT or MQPUT1 call to put a message to each of those queues

Testing the IBM MQ implementation

Figure 5-16. Distribution lists

You can use distribution lists to put a message to multiple destinations in a single MQPUT or MQPUT1 call. A single MQOPEN call can open multiple queues, and a single MQPUT call can then put a message to each of those queues.

A distribution list is a set of object records, where each object record has two fields. *ObjectName* specifies the queue name. *ObjectQMgrName* specifies the queue manager name of a single destination queue.

A distribution list might contain the name of an alias queue or the name of a local definition of a remote queue. The example of a distribution list that is shown in the figure makes the following assumptions:

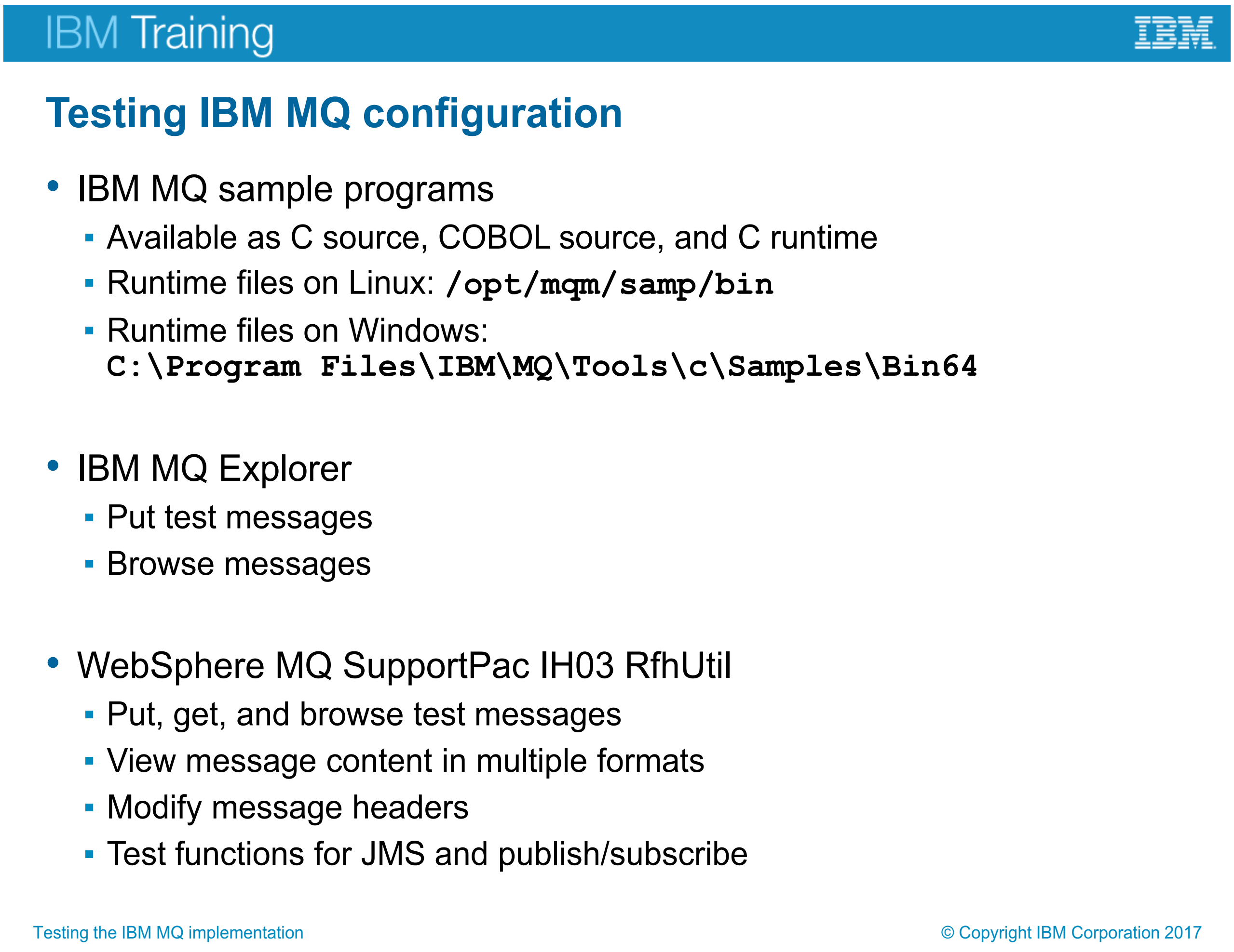
- MAIL_IN_HURSLEY is an alias queue that resolves to the local queue MAIL_IN on queue manager HURSLEY.
- MAIL_IN_PARIS is a remote queue definition that resolves to the queue MAIL_IN on queue manager PARIS by using transmission queue PARIS.XMIT.
- MAIL_IN_DALLAS is a remote queue definition that resolves to the queue MAIL_IN on queue manager DALLAS by using transmission queue DALLAS.XMIT.

After a distribution list is opened, the application can call MQPUT to put a message on the queues in the list. As a response to the call, the queue manager puts **one** copy of the message on each local queue, including the transmission queues. So, only one copy of the message is put on the transmission queue DALLAS even though the message is ultimately destined for two queues.

A distribution header and a transmission queue header prefix the application data on the transmission queue. The message that is transmitted between HURSLEY and DALLAS effectively contains a distribution list for the two destinations.

An important property of the implementation is that a message that is destined for multiple queues is replicated only at the last possible point at each stage of its journey. In this way, network traffic is minimized.

You learn more about transmission queues and queue manager communication in Unit 6, “Implementing distributed queuing”.



IBM Training

Testing IBM MQ configuration

- IBM MQ sample programs
  - Available as C source, COBOL source, and C runtime
  - Runtime files on Linux: **`/opt/mqm/samp/bin`**
  - Runtime files on Windows: **`C:\Program Files\IBM\MQ\Tools\c\Samples\Bin64`**
- IBM MQ Explorer
  - Put test messages
  - Browse messages
- WebSphere MQ SupportPac IH03 RfhUtil
  - Put, get, and browse test messages
  - View message content in multiple formats
  - Modify message headers
  - Test functions for JMS and publish/subscribe

Testing the IBM MQ implementation © Copyright IBM Corporation 2017

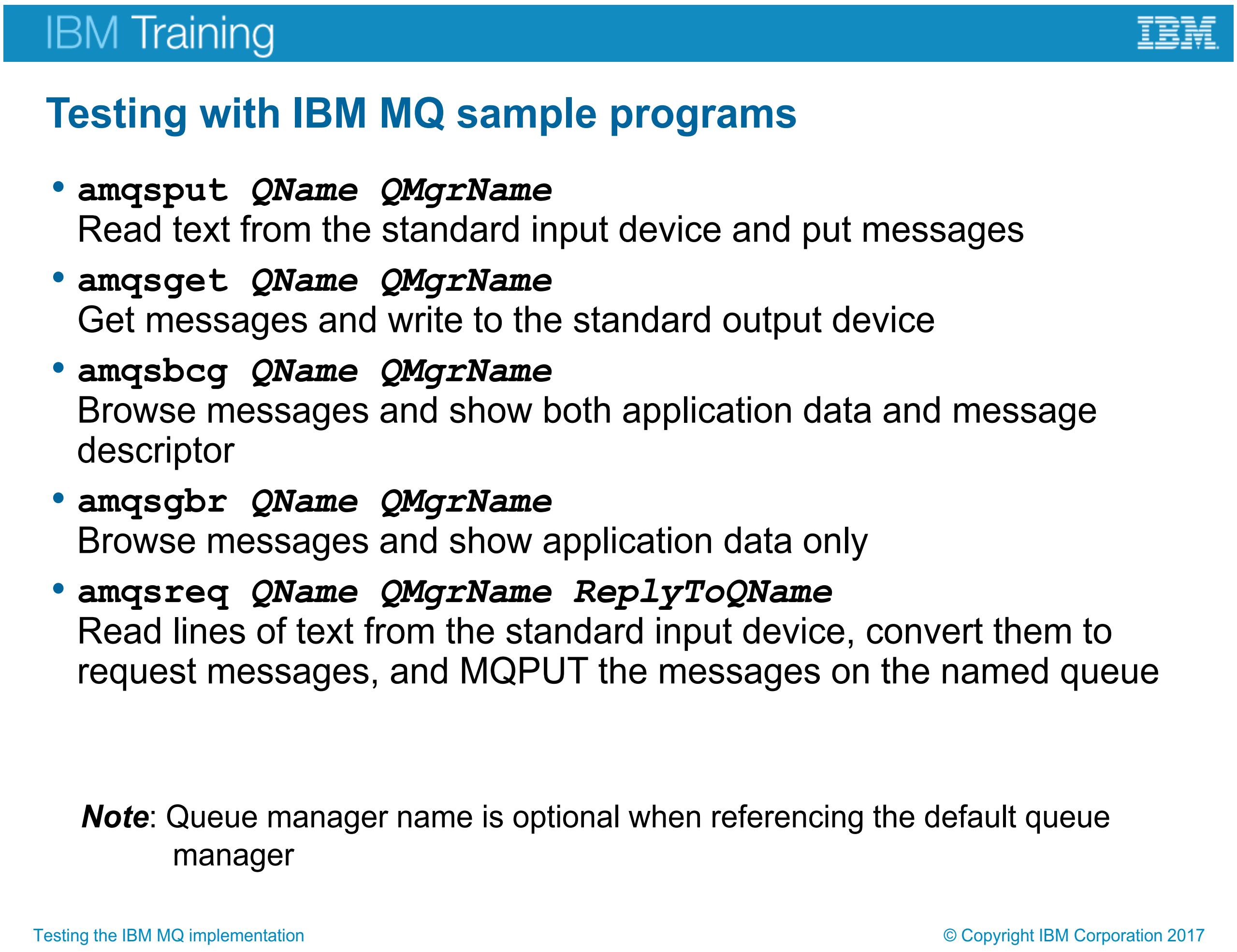
Figure 5-17. Testing IBM MQ configuration

MQ provides many ways to test your MQ configuration.

IBM MQ contains sample programs that illustrate the use of MQI calls. MQ sample programs are available as C source, COBOL source, and C runtime. They can be run from any location on Windows. On Linux and UNIX, you must run the sample program from its home directory or set the PATH to the sample program directory.

MQ Explorer also has some sample programs for putting and browsing messages.

Many MQ administrators use the WebSphere SupportPac IH03 RFHutil for testing queues. It allows test messages to be captured and stored in files. Messages can also be read and displayed in various formats. At the time of publication of this course, WebSphere SupportPac IH03 was not yet updated to support MQ V9.



IBM Training

Testing with IBM MQ sample programs

- ***amqsput** QName QMgrName*
  Read text from the standard input device and put messages
- ***amqsget** QName QMgrName*
  Get messages and write to the standard output device
- ***amqsbcg** QName QMgrName*
  Browse messages and show both application data and message descriptor
- ***amqsgbr** QName QMgrName*
  Browse messages and show application data only
- ***amqsreq** QName QMgrName ReplyToQName*
  Read lines of text from the standard input device, convert them to request messages, and MQPUT the messages on the named queue

Note: Queue manager name is optional when referencing the default queue manager

Testing the IBM MQ implementation © Copyright IBM Corporation 2017

Figure 5-18. Testing with IBM MQ sample programs

This figure describes some of the sample programs that MQ provides.

The Put sample program, **amqsput**, connects to the queue manager and opens the queue. It reads lines of text from the standard input device, generates a message from each line of text, and puts the messages on the named queue. After it reads a null line or the EOF character from the standard input device, it closes the queue, and disconnects from the queue manager.

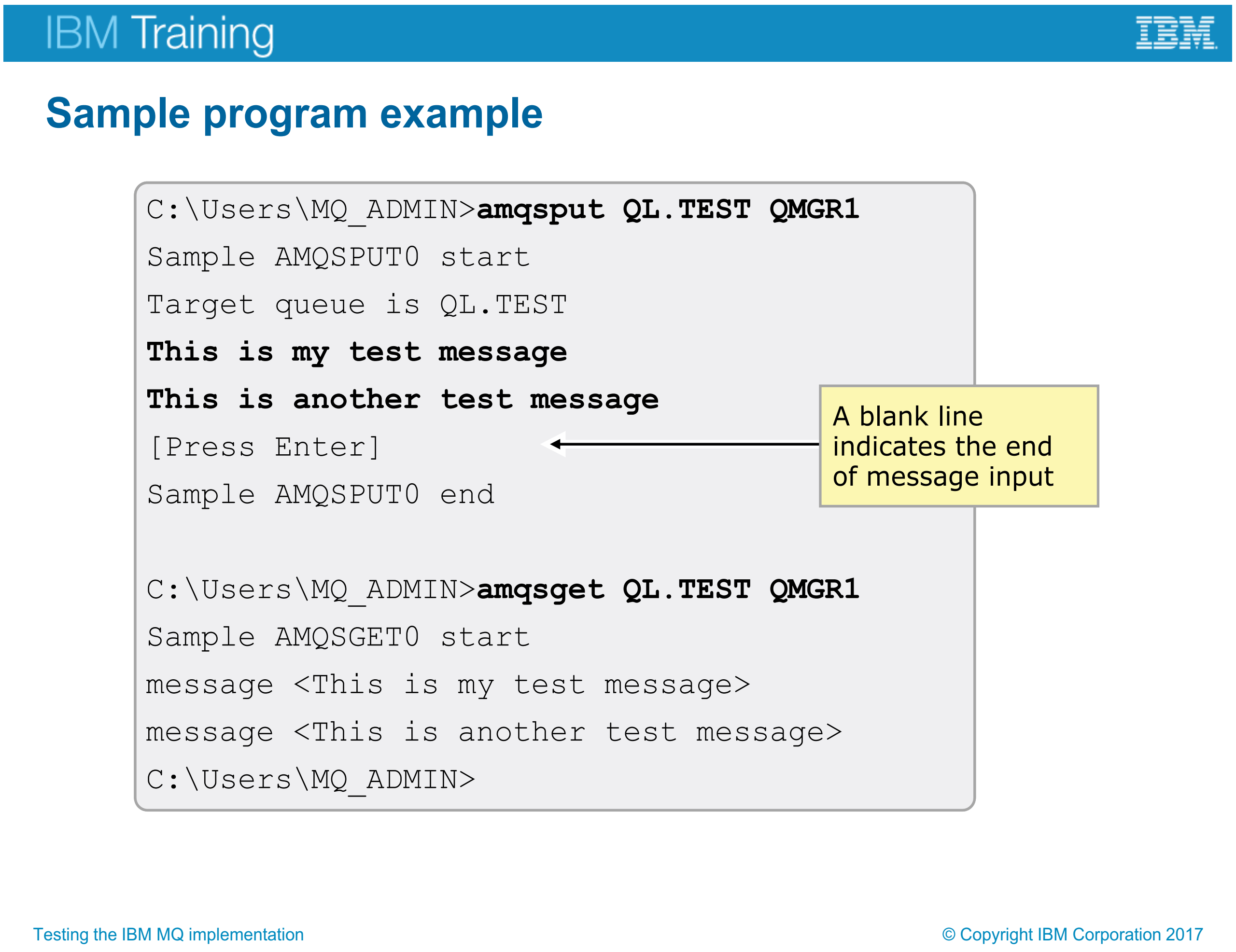
The Get sample program, **amqsget**, connects to the queue manager, opens the queue for input, and gets all the messages from the queue. It writes the text within the message to the standard output device, waits 15 seconds (60 seconds if no message exists at the start) in case any more messages are put on the queue. The program then closes the queue and disconnects from the queue manager.

MQ provides two browser sample programs:

- The **amqsbcg** program connects to the queue manager and opens the queue for browsing. It browses all the messages on the queue and writes their contents, in both hexadecimal and character format, to the standard output device. It also shows, in a readable format, the fields in the message descriptor for each message. It then closes the queue and disconnects from the queue manager.
- The **amqsbr** program browses messages on a queue by using the MQGET call.

The Request sample program, **amqsreq**, demonstrates client/server processing. It puts a series of request messages on the target server queue by using the MQPUT call. These messages specify the local queue, SYSTEM.SAMPLE.REPLY, as the reply-to queue, which can be a local or remote queue.

You use many of these sample programs in the lab exercises for this course.


IBM Training

Sample program example

```
C:\Users\MQ_ADMIN>amqsput QL.TEST QMGR1
Sample AMQSPUT0 start
Target queue is QL.TEST
This is my test message
This is another test message
[Press Enter]
Sample AMQSPUT0 end

C:\Users\MQ_ADMIN>amqsget QL.TEST QMGR1
Sample AMQSGET0 start
message <This is my test message>
message <This is another test message>
C:\Users\MQ_ADMIN>
```

A blank line indicates the end of message input

Testing the IBM MQ implementation © Copyright IBM Corporation 2017

Figure 5-19. Sample program example

This figure shows an example of the **amqsput** and **amqsget** sample programs.

The **amqsput** and **amqsget** sample programs require two parameters: the queue name and the queue manager name.

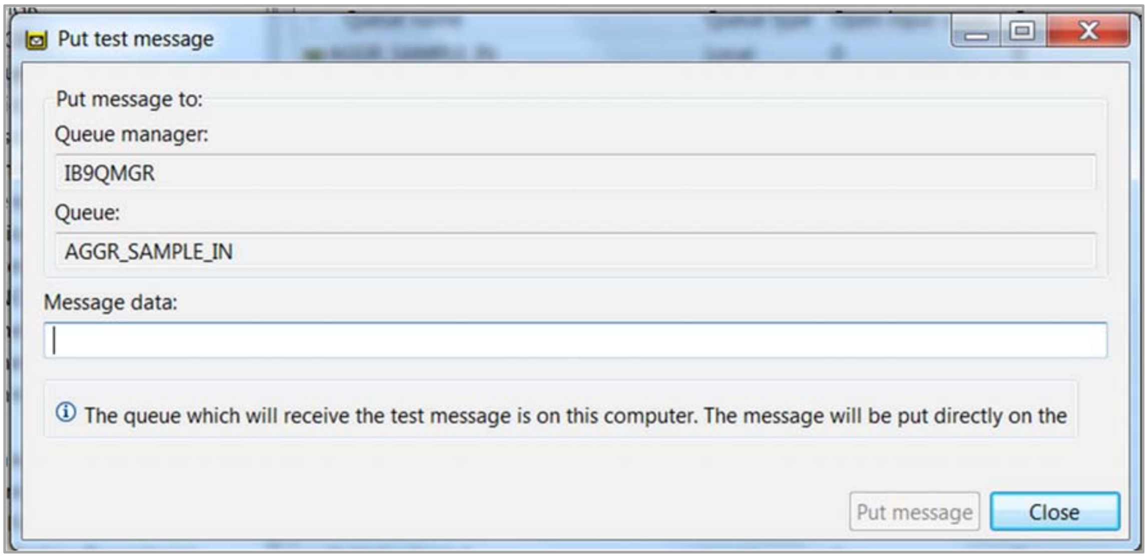
To use the **amqsput** sample program, enter one or more test messages after the program starts. When you are finished entering test messages, press Enter on a blank line to end the sample program.

As shown in the example, the **amqsget** program gets all the messages that are in the queue, which empties the queue. When you run the **amqsget** sample program, the program waits for some time in case any more messages are written to the queue; it might take a minute for the program to complete.

IBM Training

Testing with IBM MQ Explorer

- Browse messages on queue
- Clear messages on queue
- Put messages on queue
- View message contents and properties



Put message to:
Queue manager: IB9QMGR
Queue: AGGR_SAMPLE_IN
Message data:

The queue which will receive the test message is on this computer. The message will be put directly on the

Testing the IBM MQ implementation © Copyright IBM Corporation 2017

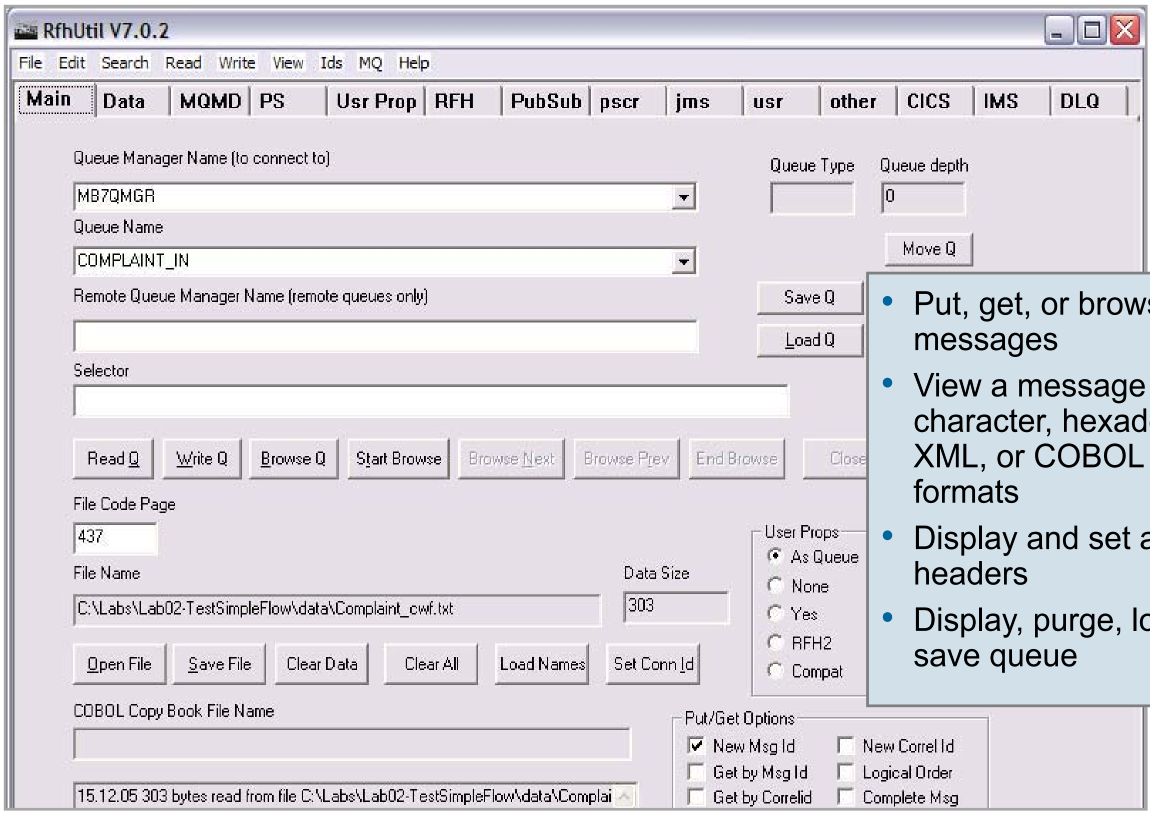
Figure 5-20. Testing with IBM MQ Explorer

With MQ Explorer, you can put messages, browse messages, clear (get) messages, and view message contents and properties.

To use the MQ Explorer test options, right-click the queue name in the **Queues** content view and then click **Test**.

IBM Training

Testing with RfhUtil (SupportPac IH03)


RfhUtil V7.0.2
File Edit Search Read Write View Ids MQ Help
Main Data MQMD PS Usr Prop RFH PubSub pscr jms usr other CICS IMS DLQ
Move Q
Save Q
Load Q
Read Q Write Q Browse Q Start Browse Browse Next Browse Prev End Browse Close
Open File Save File Clear Data Clear All Load Names Set Conn Id

Queue Manager Name (to connect to): MB7QMGR

Queue Name: COMPLAINT_IN

Remote Queue Manager Name (remote queues only):

Selector:

Queue Type:

Queue depth: 0

File Code Page: 437

File Name: C:\Labs\Lab02-TestSimpleFlow\data\Complaint_cwf.txt

Data Size: 303

User Props: [✓ As Queue] None Yes RFH2 Compat

COBOL Copy Book File Name:

Put/Get Options: [✓ New Msg Id] New Correl Id, Get by Msg Id, Logical Order, Get by Correlid, Complete Msg

```
15.12.05 303 bytes read from file C:\Labs\Lab02-TestSimpleFlow\data\Complai
```

- Put, get, or browse messages
- View a message in character, hexadecimal, XML, or COBOL formats
- Display and set all headers
- Display, purge, load, or save queue

Testing the IBM MQ implementation © Copyright IBM Corporation 2017

Figure 5-21. Testing with RFHUtil (SupportPac IH03)

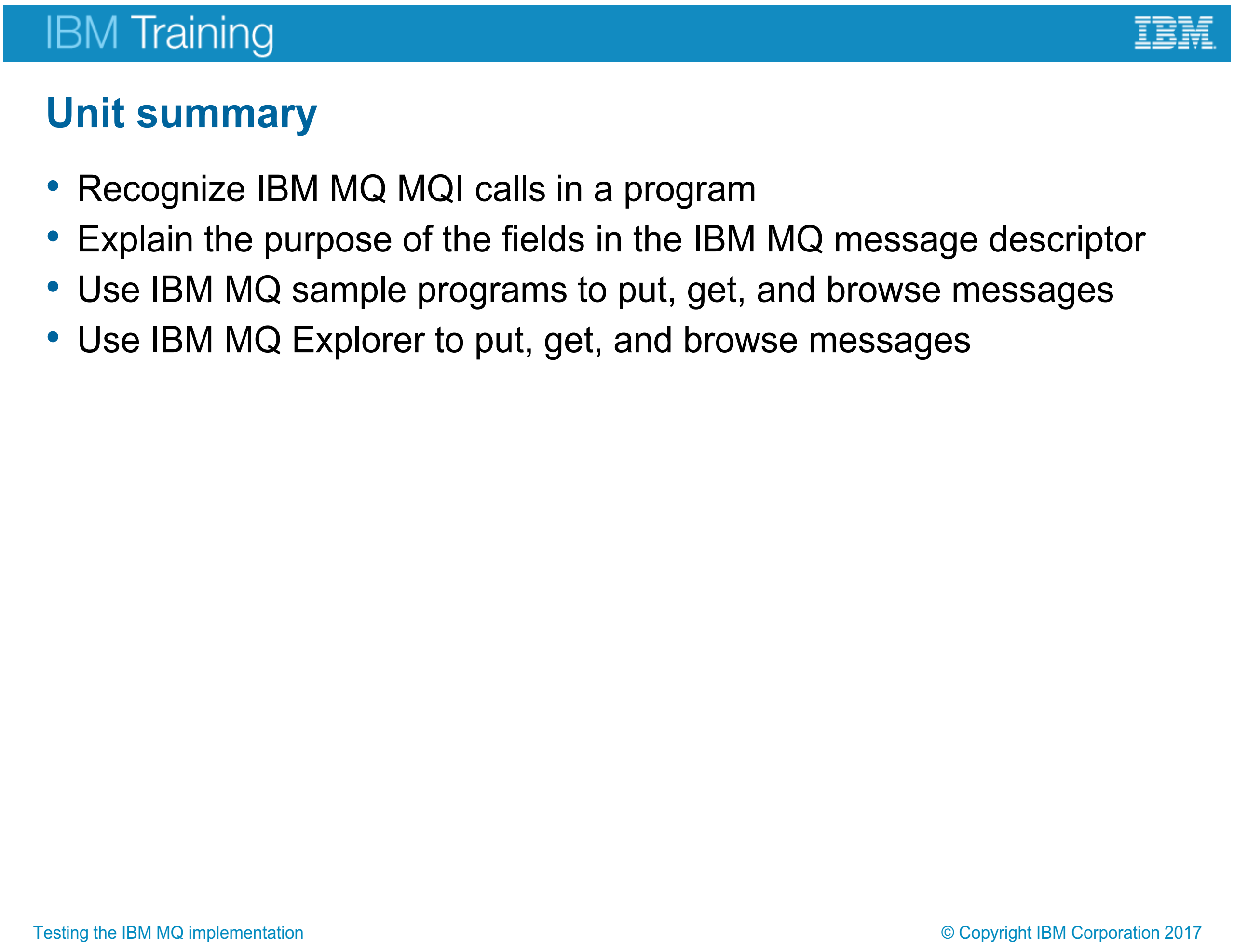
The RFHUtil utility program is a WebSphere MQ SupportPac. It reads data from files, queues, or both. It also writes data to files, queues, or both, and displays data in various formats.

The user data portion of the message can be displayed in various formats, but it cannot be changed. Another program must be used to create or change the user data.

With RFHUtil, you can add rules and formatting headers to messages or files. It writes and formats these headers when they are found in messages or files that it reads. The headers can include publish/subscribe commands.

You can download RFHUtil from the WebSphere MQ SupportPac website at: http://www.ibm.com/support

RFHutil is a category 2 SupportPac. It is free and fully supported.



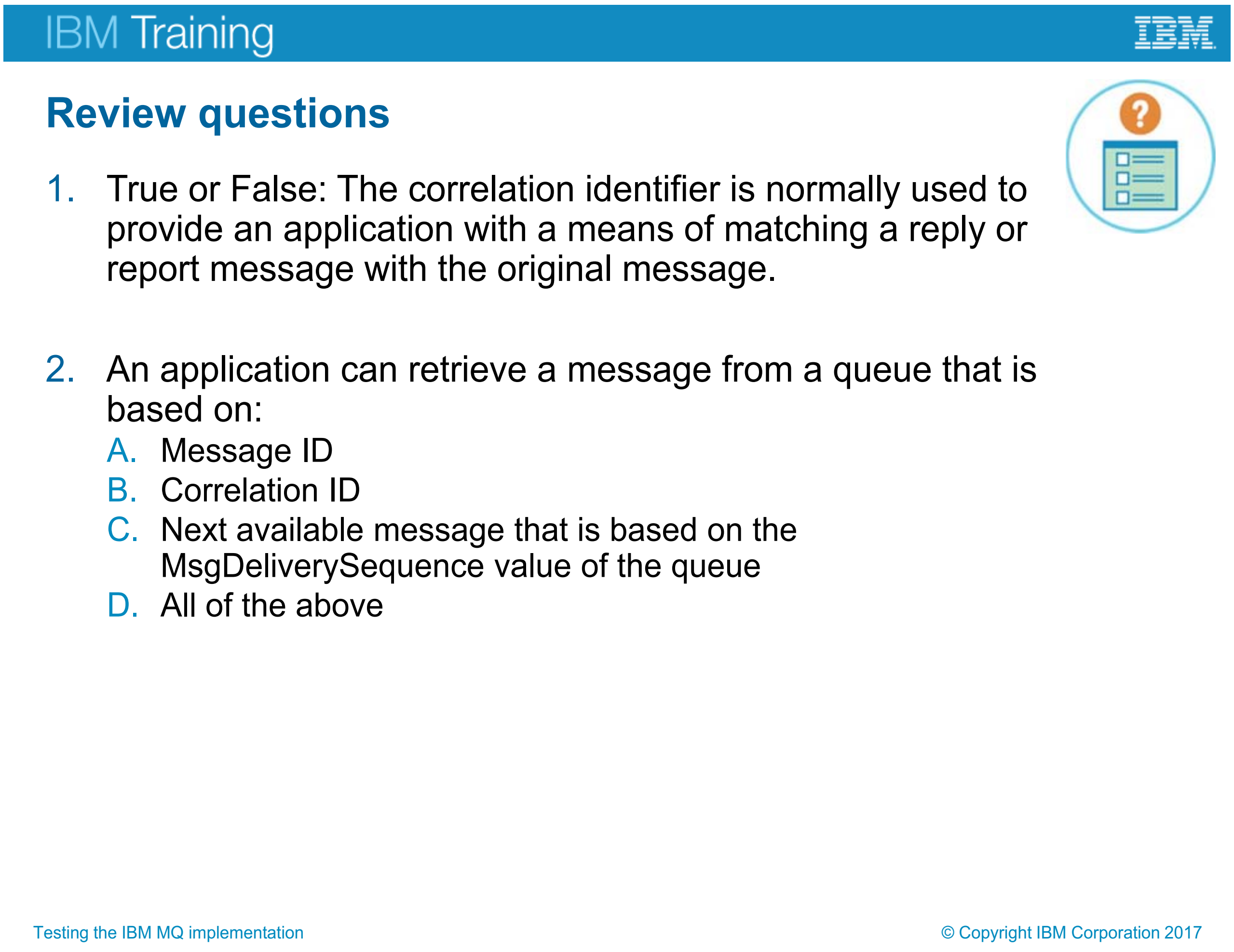
IBM Training

Unit summary

- Recognize IBM MQ MQI calls in a program
- Explain the purpose of the fields in the IBM MQ message descriptor
- Use IBM MQ sample programs to put, get, and browse messages
- Use IBM MQ Explorer to put, get, and browse messages

Testing the IBM MQ implementation © Copyright IBM Corporation 2017

Figure 5-22. Unit summary



IBM Training

IBM

Review questions

1. True or False: The correlation identifier is normally used to provide an application with a means of matching a reply or report message with the original message.

2. An application can retrieve a message from a queue that is based on:
   A. Message ID
   B. Correlation ID
   C. Next available message that is based on the MsgDeliverySequence value of the queue
   D. All of the above

Testing the IBM MQ implementation

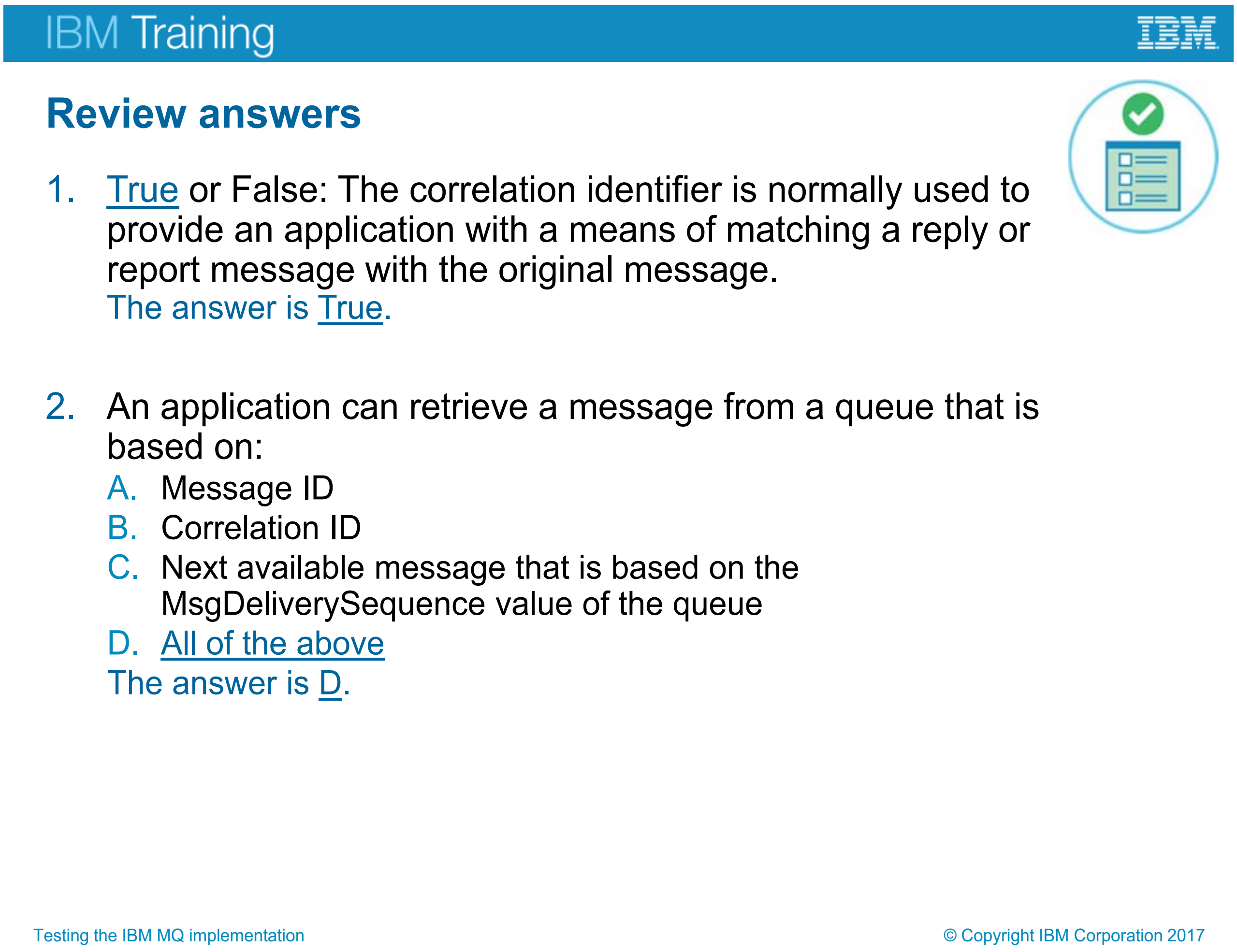
© Copyright IBM Corporation 2017

Figure 5-23. Review questions

Write your answers down here:

1.

2.



IBM Training

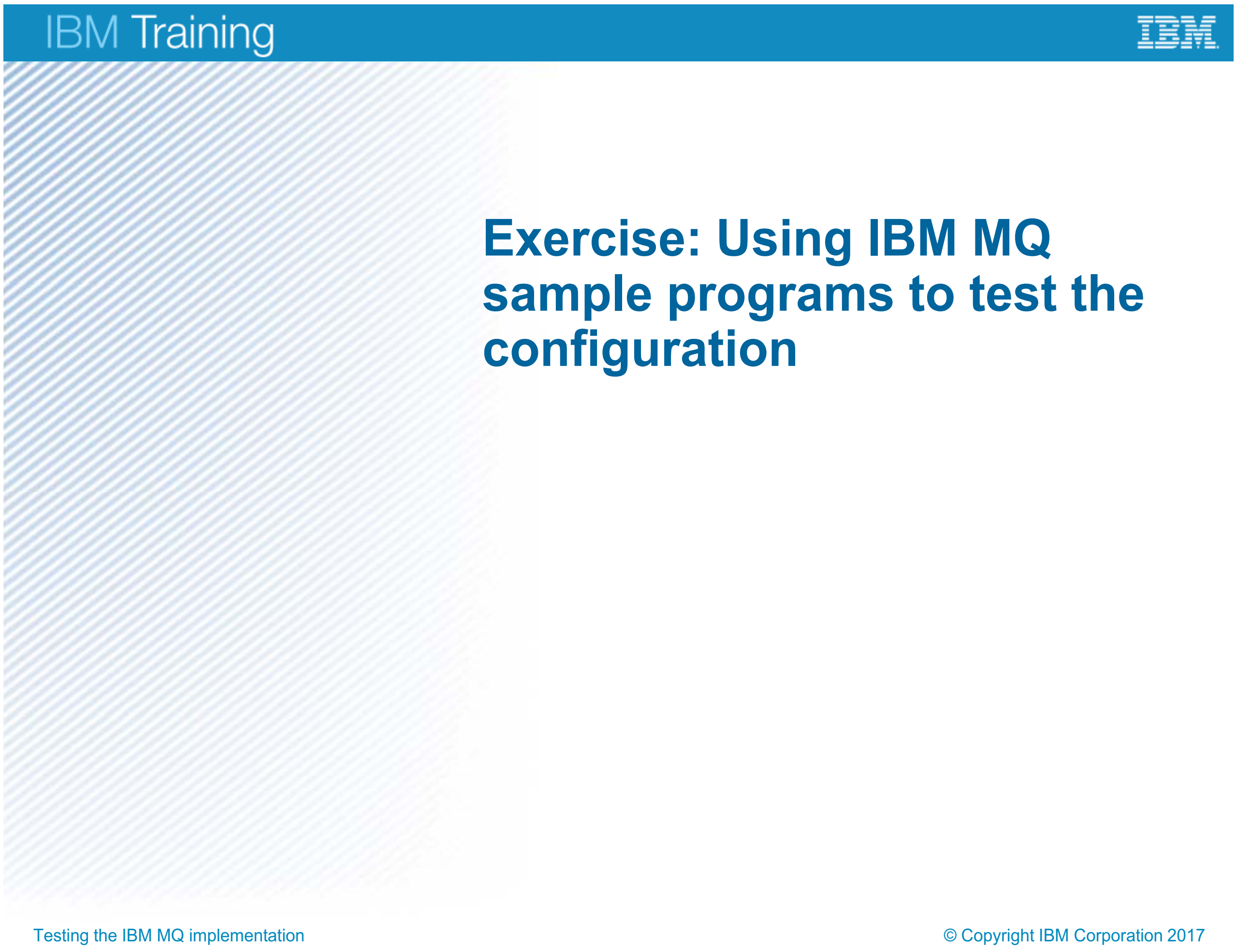
Review answers

1. True or False: The correlation identifier is normally used to provide an application with a means of matching a reply or report message with the original message.
   The answer is True.

2. An application can retrieve a message from a queue that is based on:
   A. Message ID
   B. Correlation ID
   C. Next available message that is based on the MsgDeliverySequence value of the queue
   D. All of the above
   The answer is D.

Testing the IBM MQ implementation © Copyright IBM Corporation 2017

Figure 5-24. Review answers



IBM Training

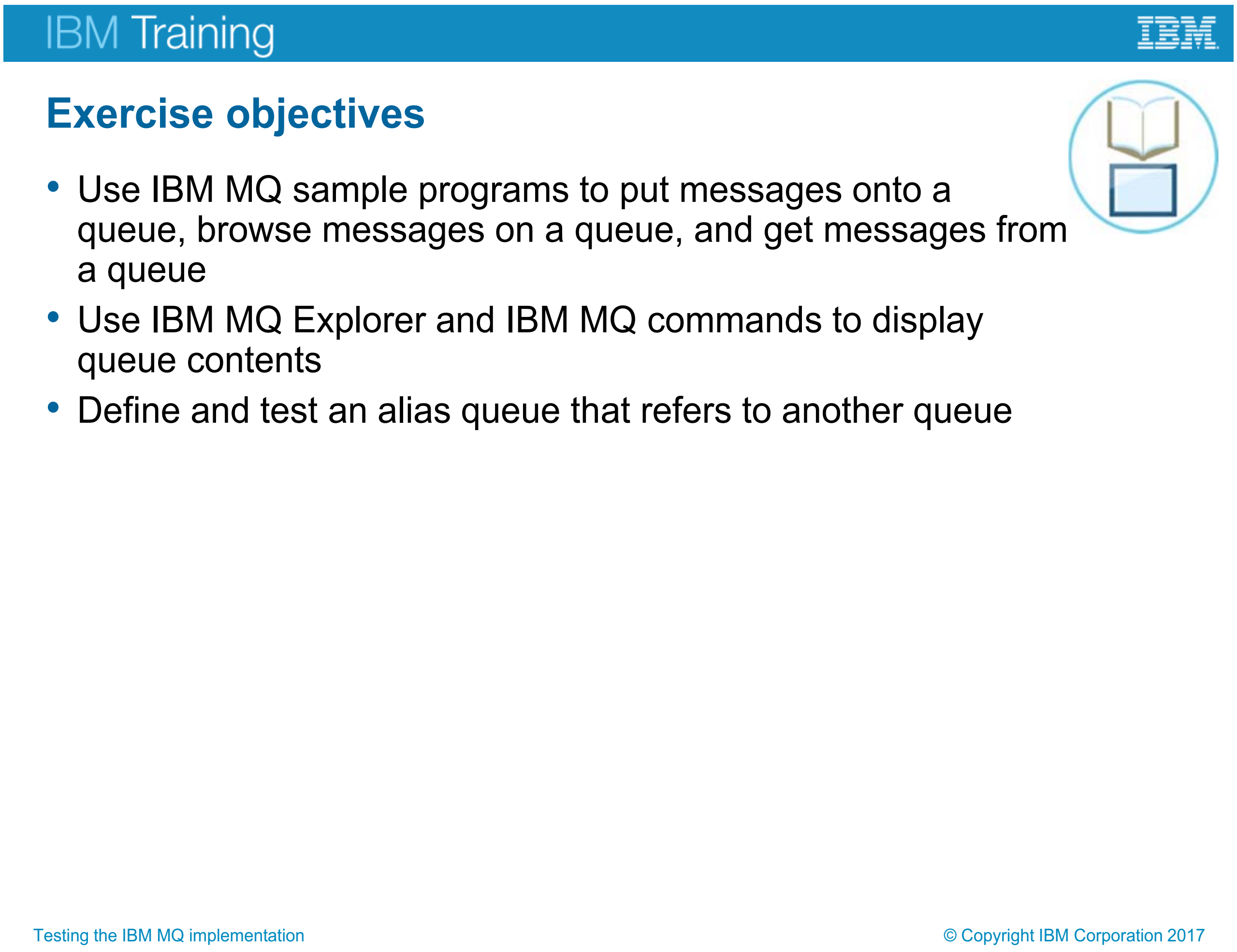
Exercise: Using IBM MQ sample programs to test the configuration

Testing the IBM MQ implementation

© Copyright IBM Corporation 2017

Figure 5-25. Exercise: Using IBM MQ sample programs to test the configuration

In this exercise, you use the IBM MQ sample programs to put, get, and browse queues and test your queue manager configuration. You then use IBM MQ Explorer and IBM MQ commands to verify the actions of the sample programs. You also define and test an alias queue.



IBM Training

Exercise objectives

- Use IBM MQ sample programs to put messages onto a queue, browse messages on a queue, and get messages from a queue
- Use IBM MQ Explorer and IBM MQ commands to display queue contents
- Define and test an alias queue that refers to another queue

Testing the IBM MQ implementation

© Copyright IBM Corporation 2017

Figure 5-26. Exercise objectives

See the *Course Exercises Guide* for detailed instructions.