Course Guide

Technical Introduction to IBM MQ

Course code WM103 / ZM103 ERC 1.0

IBM Training

July 2016 edition

Notices

This information was developed for products and services offered in the US.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml.

Contents

Trademarks viii

Course description ix

Agenda xi

Unit 1. IBM MQ overview 1-1
How to check online for course material updates 1-2
Unit objectives 1-3
The business challenge 1-4
Business drivers for enterprises 1-5
Application connections 1-6
Application connectivity options 1-7
Message-oriented middleware 1-8
Advantages of message queuing 1-9
Message-queuing styles 1-10
The power of effective messaging 1-11
IBM MQ 1-12
Benefits of IBM MQ in the enterprise 1-13
IBM MQ software compatibility 1-15
IBM MQ development options 1-16
IBM MQ development options 1-17
IBM MQ Light API 1-18
Deployment options 1-19
Using IBM MQ as a service 1-20
IBM MQ delivery and support 1-22
IBM MQ delivery and support versions 1-23
Why do customers value and trust IBM MQ? 1-24
Unit summary 1-25
Review questions 1-26
Review answers 1-27

Unit 2. IBM MQ basics 2-1
Unit objectives 2-2
IBM MQ components 2-3
Queue manager 2-4
Queue manager terminology 2-5
Messages 2-6
Message persistence 2-7
Queues 2-8
Queue attributes 2-9
Shared queues on z/OS 2-10
Special purpose local queues (1 of 2) 2-11
Special purpose local queues (2 of 2) 2-12
Terminology checkpoint 2-13
Channels 2-14
Message channel pair example: Sender-receiver 2-15
Message expiration 2-16
Report messages 2-17
Application programming interfaces 2-18

MQI overview . . . 2-19
Starting IBM MQ applications by using triggers . . . 2-20
Triggering: Process scenario . . . 2-21
Triggering: Channel scenario . . . 2-22
Transactions: Terminology . . . 2-23
Transactions: Global units of work . . . 2-24
IBM MQ client . . . 2-25
Managing client connections . . . 2-26
IBM MQ clients . . . 2-27
Unit summary . . . 2-28
Review questions . . . 2-29
Review answers . . . 2-30

Unit 3. Messaging styles, topologies, and architecture overview . . . 3-1
Unit objectives . . . 3-2
Point-to-point and publish/subscribe . . . 3-3
Point-to-point and publish/subscribe comparison . . . 3-4
Point-to-point messaging example . . . 3-5
Distributed publish/subscribe topologies . . . 3-6
What is a queue manager cluster? . . . 3-7
Queue manager cluster components . . . 3-8
Clustered queues and channels . . . 3-9
Benefits of IBM MQ clusters . . . 3-10
Highly available IBM MQ topologies . . . 3-11
Multi-instance queue managers: All is well . . . 3-12
Multi-instance queue managers: Problem occurs . . . 3-13
Multi-instance queue managers: Fail over, then reconnect . . . 3-14
IBM MQ failover technology comparison . . . 3-15
Highly scalable IBM MQ topologies . . . 3-16
Terminology checkpoint: Multi-instance and multi-version . . . 3-17
The case for standards and governance . . . 3-18
Unit summary . . . 3-20
Review questions . . . 3-21
Review answers . . . 3-22

Unit 4. System administration overview. . . . 4-1
Unit objectives . . . 4-2
Choosing an IBM MQ administration method . . . 4-3
Different administrative needs . . . 4-4
IBM MQ administrative tasks . . . 4-5
IBM MQ installation . . . 4-6
IBM MQ control commands . . . 4-7
Examples of IBM MQ control commands . . . 4-8
IBM MQ script (MQSC) commands . . . 4-9
Examples of MQSC commands . . . 4-10
Programmable Command Format (PCF) commands . . . 4-11
IBM MQ Administration Interface (MQAI) . . . 4-12
IBM MQ Explorer . . . 4-13
IBM MQ Explorer: Queue manager administration . . . 4-14
IBM MQ Explorer: Queue administration . . . 4-15
IBM MQ Explorer: Channel administration . . . 4-16
IBM MQ Explorer: Cluster administration . . . 4-17
IBM MQ Explorer: Publish/subscribe administration . . . 4-18
z/OS ISPF panels . . . 4-19
Backup and recovery . . . 4-20
Queue manager logs and recovery . . . 4-21

IBM i journals . . . 4-22
IBM MQ logging considerations . . . 4-23
Monitoring and auditing: Instrumentation events . . . 4-24
Statistics and accounting data . . . 4-25
Performance monitoring . . . 4-26
Unit summary . . . 4-27
Review questions . . . 4-28
Review answers . . . 4-29

Unit 5. Security overview . . . 5-1
Unit objectives . . . 5-2
Terminology checkpoint: Security areas . . . 5-3
Security considerations for IBM MQ . . . 5-4
Access control mechanisms . . . 5-5
Access control to administer IBM MQ . . . 5-6
Identification mechanisms: Context . . . 5-7
Security at the link and application levels . . . 5-8
Link-level security: Identification and authentication . . . 5-9
Link-level security: MCAUSER . . . 5-10
Channel blocking points . . . 5-11
Channel authentication rules . . . 5-12
Authentication mechanisms: Security exits . . . 5-13
Link-level security: Connection authentication . . . 5-14
Link-level security: Confidentiality, data integrity, and authentication . . . 5-15
Application-level security: Confidentiality and data integrity . . . 5-16
IBM MQ Advanced Message Security . . . 5-17
IBM MQ Advanced Message Security qualities of protection . . . 5-18
IBM MQ Internet Pass-Thru (MQIPT) . . . 5-19
Auditing . . . 5-20
IBM MQ security summary . . . 5-21
Unit summary . . . 5-22
Review questions . . . 5-23
Review answers . . . 5-24

Unit 6. Introduction to IBM MQ Managed File Transfer . . . 6-1
Unit objectives . . . 6-2
Introducing IBM MQ Managed File Transfer . . . 6-3
IBM MQ Managed File Transfer features . . . 6-4
Sample IBM MQ Managed File Transfer architecture . . . 6-5
Product components . . . 6-6
Product options . . . 6-7
How does it work? . . . 6-8
Required IBM MQ connectivity paths . . . 6-9
Transfer request options: Resource monitors . . . 6-10
Transfer request options: Command . . . 6-11
Transfer request options: Message on command queue . . . 6-12
Transfer request options: IBM MQ Explorer . . . 6-13
Verifying transfer with IBM MQ Explorer . . . 6-14
Auditing loggers . . . 6-15
Protocol bridge . . . 6-16
Connect:Direct bridge . . . 6-17
Unit summary . . . 6-18
Review questions . . . 6-19
Review answers . . . 6-20

Unit 7. Introduction to IBM MQ Telemetry and IBM MessageSight 7-1
Unit objectives 7-2
What is MQTT? 7-3
Publish and subscribe pattern 7-4
Some benefits of MQTT 7-5
MQTT quality of service (QoS) 7-6
MQTT use case: Mobile applications 7-7
MQTT broker implementations 7-8
IBM MQ Telemetry 7-9
IBM MQ Telemetry use case: Smart Health Services 7-10
IBM MQ Telemetry use case: Fraud detection 7-11
Benefits of IBM MQ Telemetry 7-12
IBM MQ Telemetry security 7-13
IBM MQ Telemetry administration with IBM MQ Explorer 7-14
IBM MQ Telemetry sample configuration 7-15
What is IBM MessageSight? 7-16
Edge of network messaging appliance 7-17
IBM MessageSight appliance capabilities 7-18
MessageSight architecture overview 7-19
IBM MessageSight example use cases 7-20
IBM MessageSight: Secure and reliable 7-21
IBM MessageSight web user interface 7-22
Unit summary 7-23
Review questions 7-24
Review answers 7-25

Unit 8. Introduction to the IBM MQ Appliance 8-1
Unit objectives 8-2
New challenges for enterprise messaging 8-3
Introducing IBM MQ Appliance 8-4
Why an appliance? 8-5
IBM MQ Appliance M2001 8-6
New features in the IBM MQ Appliance M2001 8-7
IBM MQ Appliance administration options 8-8
IBM MQ Console 8-9
IBM MQ Console monitoring examples 8-10
Connectivity 8-11
Is message content secure with the IBM MQ Appliance? 8-12
More about security 8-13
Disaster recovery planning 8-14
Disaster recovery 8-15
What is the value to you? 8-16
Comparison between IBM messaging appliances 8-17
Comparing IBM MQ to the IBM MQ Appliance 8-18
Can you use IBM MQ and the IBM MQ Appliance together? 8-19
Unit summary 8-20
Review questions 8-21
Review answers 8-22

Unit 9. Expanding the scope of IBM MQ. 9-1
Unit objectives 9-2
Using IBM MQ with IMS 9-3
IBM MQ IMS bridge 9-4
IBM MQ and CICS 9-5
IBM MQ CICS bridge 9-6
JMS support in IBM MQ 9-7

IBM MQ classes for JMS 9-8
IBM JMS extensions 9-9
IBM MQ resource adapter 9-10
Using IBM MQ with WebSphere Application Server 9-11
Introducing IBM Integration Bus 9-12
IBM MQ connectivity options for IBM Integration Bus 9-13
IBM Integration Bus features that require IBM MQ 9-14
IBM WebSphere Adapters 9-15
IBM MQ in the Cloud 9-16
Cloud technology checkpoint 9-17
IBM MQ Hypervisor editions 9-18
IBM MQ Advanced for PureApplication 9-19
IBM Message Hub on IBM Bluemix 9-20
IBM Message Connect on IBM Bluemix 9-21
Other Cloud options for IBM MQ 9-22
Unit summary 9-23
Review questions 9-24
Review answers 9-25

Unit 10. Course summary 10-1
Unit objectives 10-2
Course objectives 10-3
Course objectives 10-4
Course objectives 10-5
To learn more on the subject 10-6
Enhance your learning with IBM resources 10-7
Unit summary 10-8
Course completion 10-9

Appendix A. List of abbreviations A-1

Trademarks

The reader should recognize that the following terms, which appear in the content of this training document, are official trademarks of IBM or other companies:

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide.

The following are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide:

AIX®	Approach®	Bluemix®
CICS®	DB2®	GPFS™
IMS™	PowerVM®	PureApplication®
RACF®	Redbooks®	System z®
VTAM®	WebSphere®	z/OS®

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

UNIX is a registered trademark of The Open Group in the United States and other countries.

VMware and the VMware "boxes" logo and design, Virtual SMP and VMotion are registered trademarks or trademarks (the "Marks") of VMware, Inc. in the United States and/or other jurisdictions.

SoftLayer® is a trademark or registered trademark of SoftLayer, Inc., an IBM Company.

Other product and service names might be trademarks of IBM or other companies.

Course description

Technical Introduction to IBM MQ

Duration: 1 day

Purpose

In this course, you learn about IBM MQ V9 basic components and the path that messages follow when they are exchanged between applications. You also learn how IBM MQ administrative responsibilities can include the management of topic-based publish/subscribe messaging, managed file transfer, and deployments to the cloud. Topics include an overview of the support that IBM MQ provides for security, publish/subscribe, high availability, administration, logging, auditing, managed file transfer, MQTT, and cloud options.

Audience

This basic course is designed for system administrators, system architects, application developers, quality assurance specialists, and technical sales and marketing professionals.

Prerequisites

You should have skills and experience in one or more of the following specific areas:

- Communications and networking protocols
- System and network management
- Application development
- Transaction processing
- Client/server solutions

Objectives

- Summarize current business drivers and the need for flexibility
- Describe enterprise messaging and the capabilities it must provide
- Identify the main ways that IBM MQ can impact application design
- Describe the basic components of IBM MQ
- Differentiate between point-to-point and IBM MQ cluster connectivity
- Summarize queue manager and queue manager components administrative tasks
- Contrast the architectural role of IBM MQ clusters and multiple instance queue managers
- Describe the security provisions of IBM MQ and IBM MQ Advanced Message Security

- Describe how IBM MQ is used as part of the communications infrastructure to:
  - Connect application environments, such as the World Wide Web, enterprise transaction systems, and database systems
  - Manage the distribution of publisher information to appropriate subscribers
  - Provide file transfer management with IBM MQ Managed File Transfer
  - Serve as a JMS provider
  - Interface with WebSphere Application Server
  - Store in-flight messages for IBM Integration Bus
  - Interact with z/OS applications
  - Facilitate connectivity to mobile environments with IBM MQ Telemetry
- Describe the options for deployment to the Cloud

Contents

- Introduce the IBM MQ products
- Summarize support for IBM MQ in the Cloud

Curriculum relationship

This course is a prerequisite for the following courses:

- WM153, IBM MQ V9 System Administration (use Windows in labs)
- WM154, IBM MQ V9 System Administration (use Linux in labs)
- WM303, IBM WebSphere MQ V9 System Administration for z/OS
- ZM153, IBM MQ V9 System Administration (self-paced)
- WM513, IBM MQ V9 Application Development (use Windows in labs)
- WM514, IBM MQ V9 Application Development (use Linux in labs)

Agenda

Note

The following unit and exercise durations are estimates, and might not reflect every class experience.

Day 1

(00:15) Course introduction
(00:30) Unit 1. IBM MQ overview
(01:00) Unit 2. IBM MQ basics
(00:30) Unit 3. Messaging styles, topologies, and architecture overview
(00:45) Unit 4. System administration overview
(00:30) Unit 5. Security overview
(00:30) Unit 6. Introduction to IBM MQ Managed File Transfer
(00:30) Unit 7. Introduction to IBM MQ Telemetry and IBM MessageSight
(00:30) Unit 8. Introduction to the IBM MQ Appliance
(00:45) Unit 9. Expanding the scope of IBM MQ
(00:15) Unit 10. Course summary

Unit 1. IBM MQ overview

Estimated time

00:30

Overview

This unit provides an overview of the business drivers that contributed to the creation of messaging-oriented middleware (MOM) such as IBM MQ. It also covers the capabilities that are required of a MOM offering and the benefits that IBM MQ brings to the enterprise.

How you will check your progress

- Review questions

References

IBM MQ Library: www.ibm.com/software/integration/wmq/library

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: http://ibm.biz/CloudEduCourses
2. On the wiki page, locate and click the **Course Information** category.
3. Find your course in the list and then click the link.
4. The wiki page displays information for the course. If an errata document is available, it is found here.
5. If you want to download an attachment, such as an errata 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 dialog box prompts.

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

IBM Training

Unit objectives

- Describe the business drivers that demand messaging-oriented middleware
- Describe the advantages of message queuing
- List the main ways that IBM MQ can impact application design and enterprise flexibility

Figure 1-2. Unit objectives

IBM Training

The business challenge

Getting data where it needs to be in your business, reliably, securely, and quickly

- I need everything to communicate, regardless of age or language
- I need to know what happened with my data
- Parts of my business are in the Cloud
- I need to integrate with partners
- I cannot afford downtime or data loss
- Business imperatives and environments change fast

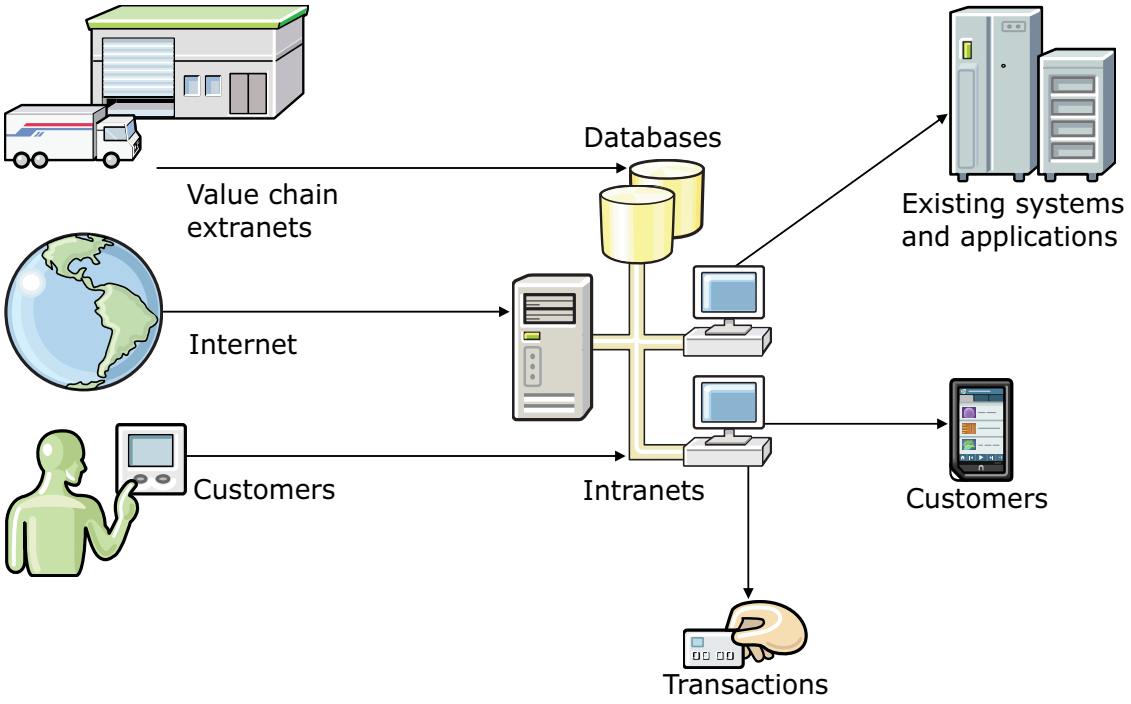
Figure 1-3. The business challenge

Businesses in a global economy face many challenges:

- The need to communicate and integrate with older systems and software and new technologies.
- The ability to track data and ensure that data is secure.
- The flexibility to support parts of the business in the Cloud.
- Providing customers with 24-hour access to their data.
- The ability to rapidly adapt to change.

IBM Training

Business drivers for enterprises



Enterprises need **infrastructure flexibility** to remain competitive

IBM MQ overview

Figure 1-4. Business drivers for enterprises

Business requirements changed in response to the “Internet of Things”. In the past, few connections across users and applications existed; today there are many.

In today’s business environment, businesses need a flexible infrastructure to remain competitive.

- Data sources in the enterprise can include value chain extranets, databases, the Internet, customers, transactions, and existing systems and applications.
- The amount of data that is exchanged between these entities is also increased, and organizations must be ready to accept high amounts of information.
- Old “batch” processes, if not disappearing, are diminishing as the requirements for real-time and on-demand information increases.
- Applications must be resilient and capable of handling failure, yet users keep expecting the same if not a better and quicker response than when they used to have a direct connection to the data.
- Customers and business partners have higher expectations, and the inability to meet the expectation results in a loss of business.

Extending existing processes would get prohibitively expensive; applications and infrastructures need to be flexible so that adapt to changing requirements.

IBM Training

Application connections

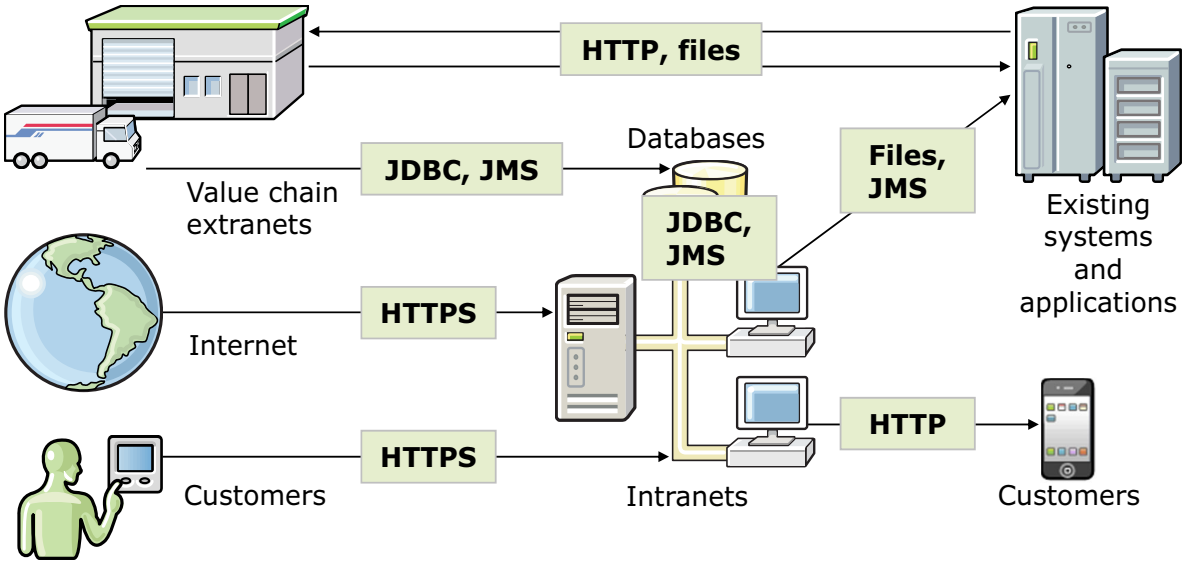


Figure 1-5. Application connections

Enterprise applications must connect to disparate data sources and targets and by using different protocols. Each of these protocols presents it own set of connectivity challenges.

- **File transfer** can be easy to set up and widely available, but no provisions exist to ensure that the data transfers completed successfully. File transfers can use various protocols such as FTP and FTPS. Also, transfers across different operating systems and countries might require data conversion.
- **HTTP** is a popular and widely available protocol for connecting to the Internet and web applications; however, it requires a tightly coupled connection. Any failing requests are lost. Also, HTTP does not have a way to save data, or to pass the data for subsequent retrieval.
- **JMS** is commonly used for transferring data to and from Java programs. The JMS API is a Java API that allows applications to create, send, receive, and read messages by using reliable, asynchronous, loosely coupled communication. A potential problem with some JMS implementations is the inability to connect to other JMS providers.

IBM Training

Application connectivity options

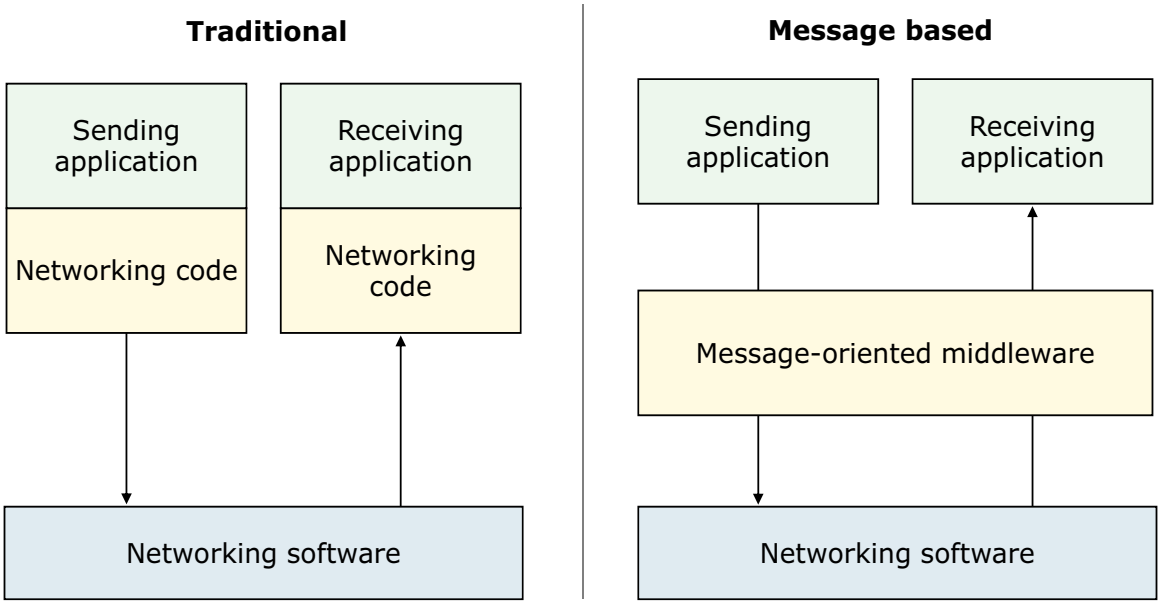


Figure 1-6. Application connectivity options

A key impediment to a flexible enterprise is traditional application connectivity, which combines business applications with one of many network application programming interfaces (APIs). This architecture requires knowledge of the networking conditions to adequately handle any communication problems. Not only does extensive work go into maintaining this code, but connections are often tightly coupled, that is, they point to a specific receiving application. Adding new connections with traditional connectivity can be labor-intensive and expensive.

In message-based application connectivity, 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.

Messaging-oriented middleware (MOM), is sometimes referred to as *enterprise messaging middleware*. It provides many options that avoid some of the limitations of transport level protocols such as HTTP and FTP.

IBM Training

Message-oriented middleware

- Software or hardware infrastructure that supports sending and receiving data in *messages* between distributed systems
  - Application modules can be distributed over heterogeneous operating systems
  - Middleware creates a distributed communications layer that insulates the application developer from the details of the various operating systems and network interfaces
- 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
- Software elements reside in all communicating components of a client/server architecture
- Typically supports asynchronous calls between the client and server applications

Figure 1-7. Message-oriented middleware

Message-oriented middleware is software or a hardware infrastructure that supports sending and receiving data in messages between distributed systems. Message-oriented middleware 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 message-oriented middleware, 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.

Software elements are present in all communicating components of a client/server architecture. Message-oriented middleware typically supports asynchronous calls between the client and server applications.

Advantages of message queuing

- Applications send data in messages through an indirect middleware layer
- Programs can process data asynchronously
- Middleware provides simple application programming interface
- Architecture of business logic is simplified

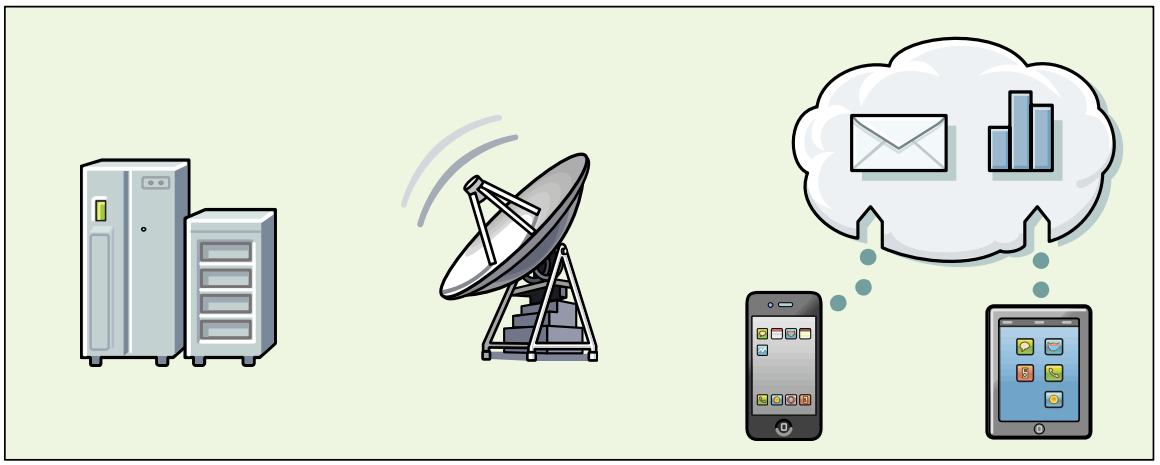


Figure 1-8. Advantages of message queuing

Applications send data in messages by use of an indirect middleware layer. In the messaging middleware, information is packaged as a *message* and sent to the receiving applications by use of a queuing system.

With message queuing, programs can communicate asynchronously; they do not need to wait for a response. Also, applications can react independently.

Middleware provides a simple application programming interface. It allows the programmer to focus on business logic and leave the network details to the middleware layer.

The architecture of business logic is simplified. It is no longer necessary to worry about numerous cases of request-reply scenarios. Also, a simpler architecture maximizes the possibility of reuse. These new simpler applications are easier to integrate into a service-oriented architecture and move to the Cloud.

IBM Training

Message-queuing styles

- Point-to-point
  - Messages are stored on a queue by a source application, and a destination application retrieves the messages
  - Application needs information about the receiving application
  - Application locates the target by using object definitions
  - Typically used when there is known to be a single producer and a single consumer of the messages

- Publish/subscribe
  - Publishing application publishes messages to an interim destination according to a topic
  - Interested recipients subscribe to the topic
  - No explicit connection between publishing and subscribing applications

Figure 1-9. Message-queuing styles

Messaging supports two primary communication methods: point-to-point and publish/subscribe.

In point-to-point messaging, a sending application must know information about the receiving application before it can send a message to that application. For example, the sending application might need to know the name of the queue to which to send the information, and might also specify a queue manager name.

In publish/subscribe messaging, a copy of each message is published by a publishing application and is delivered 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.

IBM Training

The power of effective messaging

- **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

Figure 1-10. The power of effective messaging

Effective messaging provides reliability, universal connectivity, flexibility, 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 expanded and the capacity of the system can no longer cope with the workload demand.

IBM Training

IBM MQ

- Transports any type of data as messages, enabling businesses to build flexible, reusable architectures such as service-oriented architecture (SOA) environments
- Works with a broad range of computing operating systems and hardware, 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 using or developing complex tools
- Offers a range of Qualities of Service (QoS)

Figure 1-11. IBM MQ

So what is IBM MQ?

IBM MQ is a powerful enterprise messaging that is in the market for over 25 years. It underpins a range of industry-types worldwide.

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 are written in the current popular languages.

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

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.

IBM Training

Benefits of IBM MQ in the enterprise

- Platform availability and compatibility
- Consistent API
- Administrative consistency across operating systems and hardware
- Persistent
- Secure
  - Data in transit
  - Data integrity
  - Non-repudiation
  - Data at rest
- File transfer options
- Integrated failover
- Auditing
- Supports point-to-point and publish/subscribe messaging styles
- Transactional clients
- JMS provider
- Mobility and telemetry
- Native connectivity to z/OS applications
- Options for deployment to the Cloud

Figure 1-12. Benefits of IBM MQ in the enterprise

This figure summarizes the benefits of MQ in the enterprise.

One of the major benefits of MQ is platform availability and compatibility: MQ is available on many operating systems and server hardware. In addition, all versions and installations of MQ can communicate with all other versions and implementations of MQ. Also, MQ handles data conversion across disparate operating systems, from ASCII to EBCDIC, and Big Endian to Little Endian.

The MQ Message Queue Interface (MQI) is a consistent API that is the same in all languages, and versions. When you implement a new release or FixPack for MQ, your applications continue to work.

MQ administrative consistency means that you can apply and use the administration tools to manage MQ across all supported operating systems.

MQ can store in-flight messages so that they can persist during a system restart; messages do not get lost.

With MQ, message exchange can be done in a secure manner, which eliminates the requirement to develop an in-house solution.

MQ provides robust transactional integrity and assured message delivery.

With MQ, organizations have the flexibility to design applications that send data directly, or use a publish/subscribe messaging style.

MQ is ready for the “Internet of Things,” with links to mobile applications, Cloud, file exchange, web services, and JMS.

IBM Training

IBM MQ software compatibility

- Operating systems
  - AIX
  - z/OS
  - IBM i
  - HP-UX
  - Linux: Red Hat, SUSE, Ubuntu
  - Solaris
  - Windows
- Administrative options
  - Control commands
  - IBM MQ Explorer: Eclipse-based graphical user interface
  - Programmable Command Format (PCF) messages
  - IBM MQ Script Commands (MQSC): Human-readable command interface
  - Web user interface for monitoring the IBM MQ Appliance

Figure 1-13. IBM MQ software compatibility

With MQ, you are not restricted to just one operating system. MQ is supported the most common operating systems: AIX, z/OS, IBM i, HP-UX, Linux, Solaris, and Windows.

MQ also supports many options for administration for flexible administration.

You learn more about these administration options throughout this course.

IBM Training

IBM MQ development options

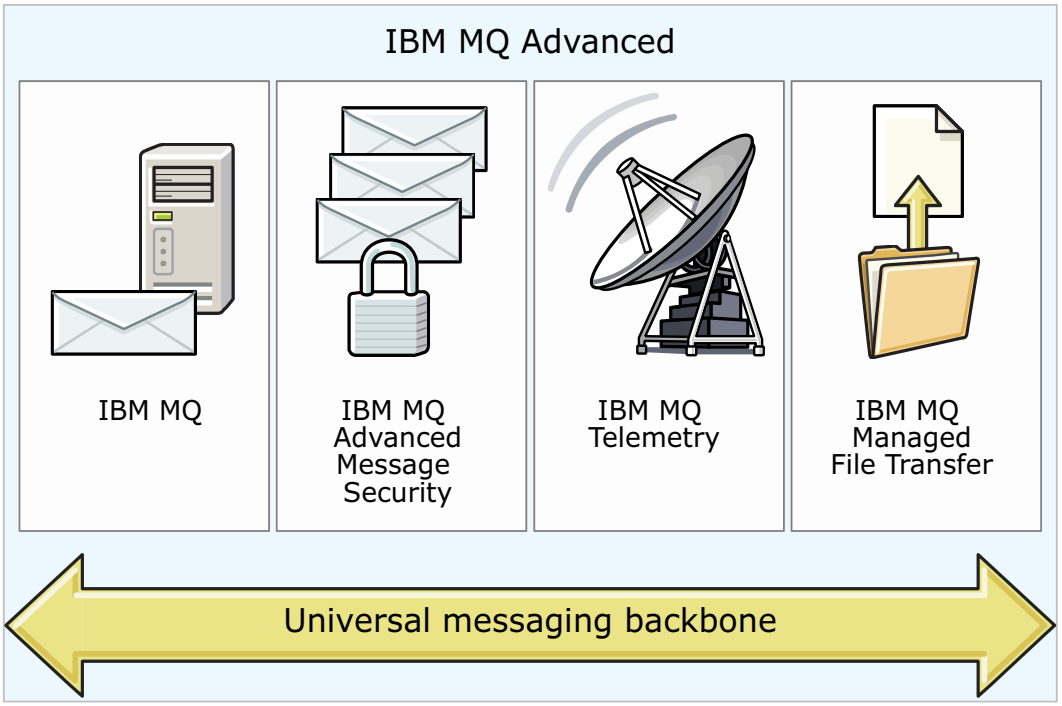


Figure 1-14. IBM MQ development options

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 development options

- IBM MQ
  - Implement a single messaging backbone that connects applications across all parts of the business, from mainframe to mobile, and on premises to cloud
  - Reduce development and administration time with simple tools
- IBM MQ Advanced
  - Strengths of 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 Advanced for Developers
  - Buy licenses of MQ at a lower cost for developers to use on their own PCs with IBM support
  - Boost developer skills in a flexible and highly configurable messaging environment
  - Faster access to all of IBM MQ Advanced capabilities for developers to enhance applications with message security, managed file transfer, and mobile access

Figure 1-15. IBM MQ development options

This figure summarizes the three 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 simplify the creation and maintenance of business applications.
- **IBM MQ Advanced** is a comprehensive messaging strategy that includes managed file transfer, advanced message security. 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 Advanced for Developers** is an option that allows you to buy licenses of IBM MQ at a lower cost for developers to use on their own workstations. It provides faster access to all of IBM MQ Advanced capabilities for developers to enhance applications with message security, managed file transfer, and mobile access.

  IBM MQ Advanced for Developers is available at no-charge for development purposes for Windows and Linux. A no-charge, 90-day trial is available for all supported operating systems.

IBM Training

IBM MQ Light API

- Makes it simple for developers to create responsive applications that are easy to scale
  - Built on AMQP
  - Developers can code in various languages, choosing from the application programming interfaces (APIs) that IBM MQ Light supplies.
- Microservices framework creates services that are independently deployable and scalable, which simplifies maintenance and provides flexibility
- Intuitive user interface assists developers by helping them to understand how applications are behaving during development
- Helps line-of-business developers making technology decisions
- Provides deployment in different IBM messaging offerings so that you can choose the appropriate messaging server without rewriting application code

Figure 1-16. IBM MQ Light API

An option for developing MQ solutions is the IBM MQ Light API.

MQ Light is a messaging API that provides a messaging run time. Developers can install, configure, and use the MQ Light API to write applications in just a few minutes, enabling developers to quickly create scalable and responsive applications.

MQ Light is built on the Advanced Message Queuing Protocol (AMQP). AMQP is an open standard application layer protocol for message-oriented middleware. The defining features of AMQP are message orientation, queuing, routing (including point-to-point and publish-and-subscribe), reliability, and security. AMQP makes it simple for developers to create responsive applications that are easy to scale.

With the MQ Light API, developers can code in various languages, choosing from the APIs that IBM MQ Light supplies: node.js, Ruby, Python, and Java.

The microservices framework that MQ Light uses means that services are independently deployable and scalable, which simplifies maintenance and provides flexibility.

The intuitive user interface assists developers by helping them to understand how applications are behaving during development. These applications can then be deployed into the IBM Bluemix cloud, or larger MQ-based infrastructures.

IBM MQ Light is available by electronic delivery only.

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
  - 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

Figure 1-17. 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

Using IBM MQ as a service

- Apply the principles of a service delivery model to an MQ deployment
- Implement self-service portals that allow lines of business professionals and application developers to request changes to the messaging environment
- Advantages:
  - Increased agility
  - Improved usability
  - Improved efficiency
  - Improved consistency

- For more information, see the IBM Redpaper “IBM MQ as a Service: A Practical Approach” at http://ibm.biz/mqaas_red

Figure 1-18. Using IBM MQ as a service

Another deployment option with MQ, is to apply service methodologies to an IBM MQ environment.

Traditionally, MQ is administered exclusively within an enterprise by a central messaging middleware team. As the number and complexity of requests increases, it can become less practical for a central team to respond within a timescale that is acceptable to the business units. An option is for enterprises to use these capabilities to implement self-service portals that allow business professionals and developers to request changes to the messaging infrastructure.

It is not essential for an enterprise to offer MQ as a service, but doing so can provide the following benefits:

- Increased agility: Developers and business professionals are empowered to provision or update the messaging resources that they require, which reduces time to value by eliminating the dependency on manual activities.
- Improved usability: Developers and business professionals can use self-service interfaces to focus on their core requirements, such as what destinations they need and what quality of service is necessary.
- Improved efficiency: Automation allows systems and messaging resources to be dynamically provisioned for test and development purposes and to simplify the promotion of changes from development through production environments.

- Improved consistency: A self-service interface provides a single entry point for changes to the messaging configuration. This interface can use common routines and workflows to deploy the necessary changes, which enforce standards and consistency.

You learn more about MQ as a service later in this course.

You can also learn more about MQ as a service in the IBM Redbooks “IBM MQ as a Service A Practical Approach”.

IBM Training

IBM MQ delivery and support

- Accelerate speed of application development
  - Developers can provision and manage their own enterprise MQ resources
  - Administrators retain overall control and confidence in the resiliency of the MQ system
- Respond faster to change by removing impact on applications and MQ teams when dynamically scaling up and scaling down deployed MQ architectures across hybrid infrastructures

Figure 1-19. IBM MQ delivery and support

To accelerate the speed of application development, IBM MQ Version 9.0 introduces a new continuous delivery and support model.

This new delivery model allows developers to access the latest 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.

IBM Training

IBM MQ delivery and support versions

- Long-Term Support Release
  - Intended for systems that require a 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 latest 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

Figure 1-20. IBM MQ delivery and support versions

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 Releases version delivers new functional enhancements, and fixes and security updates, on a much shorter release cycle. This version provides rapid access to new functions. This version is intended for systems where applications want to use the latest capabilities of IBM MQ.

For more information about the MQ release versions, see the “IBM MQ FAQ for Long-Term Support and Continuous releases” web page.

IBM Training

Why do customers value and trust IBM MQ?

- Provides a reliable solution for bridging the components in a service-oriented architecture
- Connects various IT systems across a wide range of hardware and operating systems
- Builds on 20+ years of messaging expertise and continued IBM investment
- Helps you share and exchange critical business information with ease, confidence, and security

Figure 1-21. Why do customers value and trust IBM MQ?

Customers value and trust IBM MQ because it provides a reliable solution for bridging the components in a service-oriented architecture.

MQ connects various IT systems across a wide range of hardware and operating systems, which includes AIX, z/OS, IBM i, and many versions of Linux and Windows.With MQ, you can use your existing infrastructure and system administrator skills.

MQ builds on 20+ years of messaging expertise and continued IBM investment. It is widely used and continuously evolving to take advantage of new technologies.

MQ helps you share and exchange critical business information with ease, confidence, and security.

IBM Training

Unit summary

- Describe the business drivers that demand messaging-oriented middleware
- Describe the advantages of message queuing
- List the main ways that IBM MQ can impact application design and enterprise flexibility

Figure 1-22. Unit summary

IBM Training

Review questions

1. True or False: When using message-oriented middleware, network logic is combined with application logic.

2. Which of the following attributes are true of message-oriented middleware?
   A. Middleware provides a simpler programming interface.
   B. Programs can be asynchronous, which allows applications to react independently.
   C. Messages are automatically encrypted to protect sensitive data.
   D. Business architecture is simplified.

Figure 1-23. Review questions

Write your answers here:

1.

2.

IBM Training IBM

Review answers

1. True or False: When using message-oriented middleware, network logic is combined with application logic.
   The answer is False. An advantage of message-oriented middleware is that it separates network logic from application logic. Message-oriented middleware provides a dedicated middleware layer for handling connectivity.

2. Which of the following attributes are true of message-oriented middleware?
   A. Middleware provides a simpler programming interface.
   B. Programs can be asynchronous, which allows applications to react independently.
   C. Messages are automatically encrypted to protect sensitive data.
   D. Business architecture is simplified.

   The answer is A, B, and D.

Figure 1-24. Review answers

Unit 2. IBM MQ basics

Estimated time

01:00

Overview

This unit describes the most commonly used IBM MQ components and how these components interact to exchange messages with local and remote applications.

How you will check your progress

- Review questions

References

IBM MQ Library: www.ibm.com/software/integration/wmq/library

IBM Training

Unit objectives

- Describe the components of IBM MQ
- Describe the role of a channel in distributed queuing
- Describe how a message gets from source to target queue and where messages are at different times in the process
- Summarize process triggering
- Summarize options for transaction handling
- Differentiate between an IBM MQ server and an IBM MQ client
- Describe the IBM MQ extended transactional client

Figure 2-1. Unit objectives

This unit introduces the IBM MQ basics. Business users, administrators, and architects who work with MQ in any capacity must understand these concepts to understand architectural, testing, and troubleshooting activities.

IBM MQ components

- IBM MQ components
  - Queue manager
  - Messages
  - Queues
  - Channels
- IBM MQ can be installed with MQ server software or MQ client software

  Note: In this course, the information pertains to installations of MQ server unless a topic is identified as applying to the MQ client

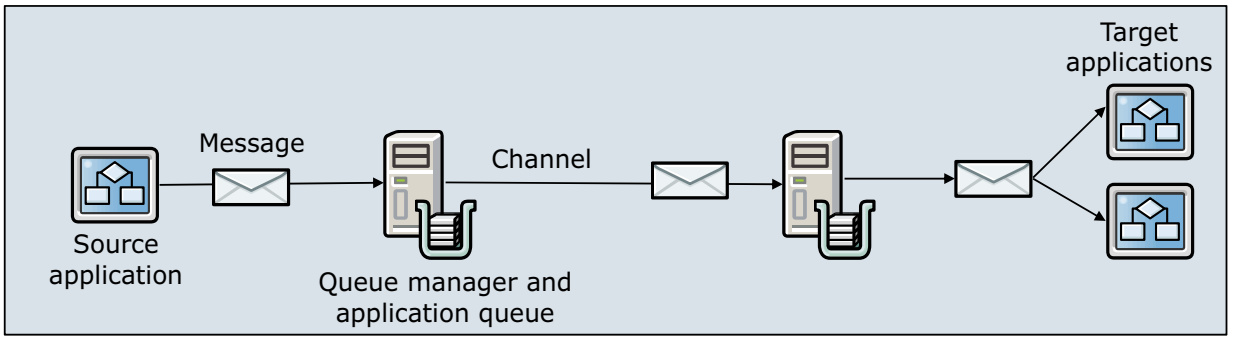


Figure 2-2. IBM MQ components

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.

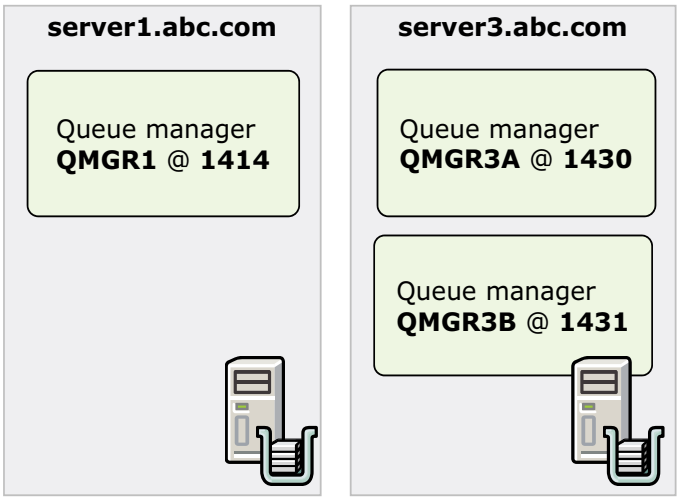
The computer that contains the IBM MQ configuration is controlled by an MQ server. Computers that contain client applications that must connect to the MQ server must also have an MQ client installation. It is possible to have both an MQ server and an MQ client installation on the same computer.

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

Queue manager

- Responsible 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
- Provides services that are needed for messaging
- Require a name and TCP/IP listener port
- A server can host more than one queue manager
  - Queue managers that share a server require different listener ports and names



IBM MQ basics

Figure 2-3. Queue manager

Queue managers own and control the objects and processes that support MQ.

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.

Each queue manager on a server is given a unique name. More than one queue manager can be defined on a server, but each queue manager must have a different name.

The queue manager communicates by using a TCP connection. A TCP connection requires a listener program at the receiving end. 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 name and port number.

In the example, the server that named is `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”

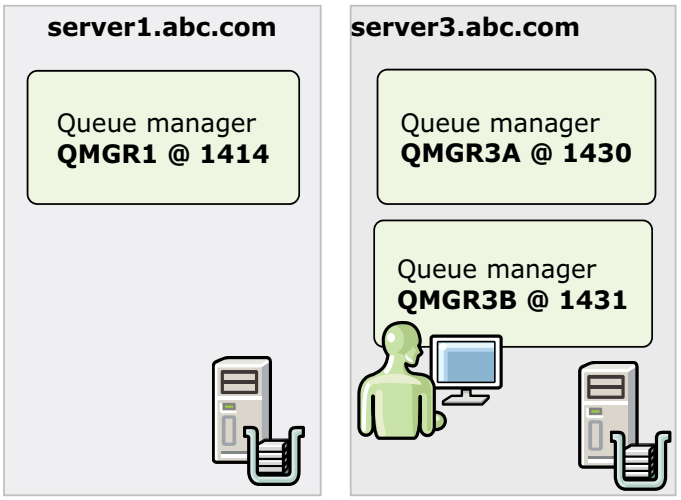


Figure 2-4. Queue manager terminology

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 you are 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.

Messages

- Messages are a distinct string of bytes exchanged between applications
- All messages contain an MQ *message descriptor* (MQMD) that identifies the message and contains control information
  Example MQMD fields: Priority, Persistence, Expiry, Encoding, ReplyToQ
- Optionally, applications can add *message properties* to a message
- *Message data* portion contains the application data
  - Application program defines the structure of the message data

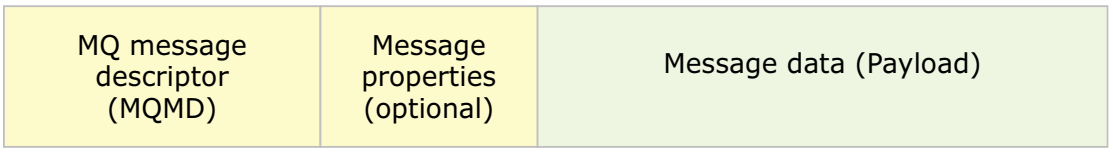


Figure 2-5. Messages

The application data that MQ exchanges is sent in the form of *messages*.

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

From your applications, you use the Message Queue Interface (MQI) to control objects. Objects are identified by the MQOD 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 determines the contents of the message data portion of the message.

IBM Training

Message persistence

- Can be defined by application in MQ message descriptor
- Persistent messages
  - Queue manager keeps a failure-tolerant recovery log of all actions that are done upon the messages
  - Recovered from the logged data after a queue manager restart
  - Use for critical business data that must be reliably maintained and not lost in a failure
- Non-persistent messages
  - Storage in system memory
  - Discarded if a queue manager stops, whether the stoppage is as a result of an operator command or because of the failure of some part of the system
  - On z/OS, stored in a coupling facility (CF) and persist while the CF remains available
  - Use where the loss of the data is not crucial because the query can be repeated and in cases where performance is considered more important than data integrity

Figure 2-6. Message persistence

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

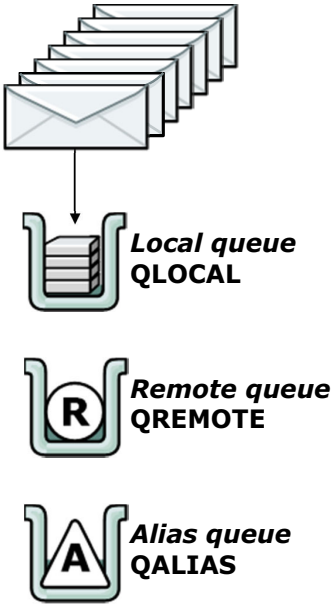
If the application identifies the message as persistent, the queue manager keeps a failure-tolerant recovery log of all actions on that message. If a queue manager restarts before the message is delivered to its target, the message is recovered from the logged data. An application should identify a message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

If the application identifies the message as non-persistent, 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 non-persistent when the loss of data is not crucial.

IBM Training

Queues

- Defined endpoint destination for messages
- Known to an application as a local queue (QLOCAL) when it is owned by the queue manager to which the application is connected
- Applications can get messages from, and put messages on local queues by using the Message Queue Interface (MQI)
- Other queue types:
  - Remote queue is a local definition that points to a queue on a remote queue manager
  - Alias queue is a pointer to a local queue or a local definition of a remote queue



Only local queues that are defined as a QLOCAL queue type hold messages

Figure 2-7. 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 MQI.

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

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 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.

IBM Training

Queue attributes

- Determine queue properties, such as maximum queue depth, maximum message length, and whether put or get operations for the queue are allowed
- Some queue attributes apply to all types of queue; other queue attributes apply only to certain types of queue
- Some attributes have default values
- Some attributes can have initial values that can be supplied by:
  - COBOL copybooks
  - C “include” files
  - IBM supplied object definitions

Figure 2-8. Queue attributes

Each queue has a set of attributes that identify the type and properties of the queue. For example, queues have a **Put inhibited** and **Get inhibited** property that control the behavior of the queue.

Some queue attributes, such as the queue type, are specified when the queue is defined, and cannot be changed. Other queue attributes of queues can be changed at any time.

Some queue attributes, such as the maximum number of messages that a queue can hold, can be changed by the queue manager during the processing of the queue. Other queue attributes, such as the **Description** attribute, can be changed by a command only. Other queue attributes, such as **Put inhibited**, can be changed by the application.

Shared queues on z/OS

- Type of local queue available on z/OS
- Queue managers in the same z/OS sysplex can access the same shared queues without extra connectivity considerations
- Several z/OS queue managers that share a queue are called a *queue sharing group*

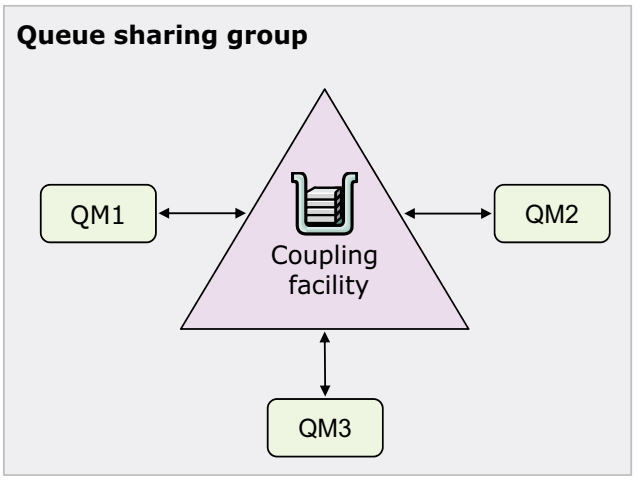


Figure 2-9. Shared queues on z/OS

On z/OS, local queued can be defined as *shared queues*.

The messages on a shared queue can be accessed by one or more queue managers that are in a sysplex. The queue managers that can access the same set of shared queues form a group that is called a *queue sharing group*.

With a queue sharing group, you can put a message on to a shared queue on one queue manager, and get the same message from the queue from a different queue manager. A shared queue provides a rapid mechanism for communication within a queue-sharing group that does not require active channels between queue managers.

IBM Training

Special purpose local queues (1 of 2)

- Transmission queue (XMITQ)
  - If messages are destined for a remote queue, the local queue manager holds them on a *transmission queue*, which persists them in a message store, until they can be forwarded to the remote queue manager
- Initiation queues
  - MQ *triggering* facility enables an application to be started automatically when messages are available to retrieve
  - If triggering is enabled for a queue and a trigger event occurs, the queue manager sends a trigger message to an initiation queue

Figure 2-10. Special purpose local queues (1 of 2)

MQ uses some local queues for specific purposes.

Transmission queues are local queues that temporarily store messages that are destined for a remote queue manager. You must define at least one transmission queue for each remote queue manager to which the local queue manager is to send messages directly. Each queue manager can have a default transmission queue.

Initiation queues are local queues that are used in triggering. A queue manager puts a trigger message on an initiation queue when a *trigger event* occurs. A trigger event is a logical combination of conditions that is detected by a queue manager. For example, a trigger event might be generated when the number of messages on a queue reaches a predefined depth. This event causes the queue manager to put a trigger message on a specified initiation queue. If a queue manager is to use triggering, at least one initiation queue must be defined for that queue manager.

IBM Training

Special purpose local queues (2 of 2)

- SYSTEM queues
  - Used by MQ for queue manager and channel operation
  - Can be browsed but unless otherwise documented must not be used to put messages

  Example:
  SYSTEM.ADMIN.STATISTICS.QUEUE that holds statistics monitoring data

- Dead-letter queue
  - Local queue that holds undeliverable messages
  - Contains the reason that a message was placed in the dead-letter queue and other pertinent information in special message header
  - Can use SYSTEM.DEAD.LETTER.QUEUE or any local queue

Figure 2-11. Special purpose local queues (2 of 2)

When a queue manager is created, several queues are created for the use of the queue manager. These queues start with the `SYSTEM` prefix and should not be changed, deleted, or used to send messages. For example, the SYSTEM.ADMIN.STATISTICS.QUEUE holds queue manager statistics data.

Another important queue is the *dead-letter queue*. The dead-letter queue is a local queue that is identified to the queue manager as the collector of messages that are in some way undeliverable.

The local queue that is used as the dead-letter queue must be identified to the queue manager. One of the SYSTEM queues is called the SYSTEM.DEAD.LETTER.QUEUE. This queue is often designated to be the dead-letter queue; however, any other local queue can serve this purpose if it is identified as the dead-letter queue to the queue manager.

IBM Training

Terminology checkpoint

Current queue manager is QMGR1

- BILL.IN is a local queue that QMGR1 owns
- SALE.IN is a remotely-owned local queue on QMGRA
- REQ.OUT is a local queue definition on QMGR1 that points to the remote queue SALE.IN on QMGR3A

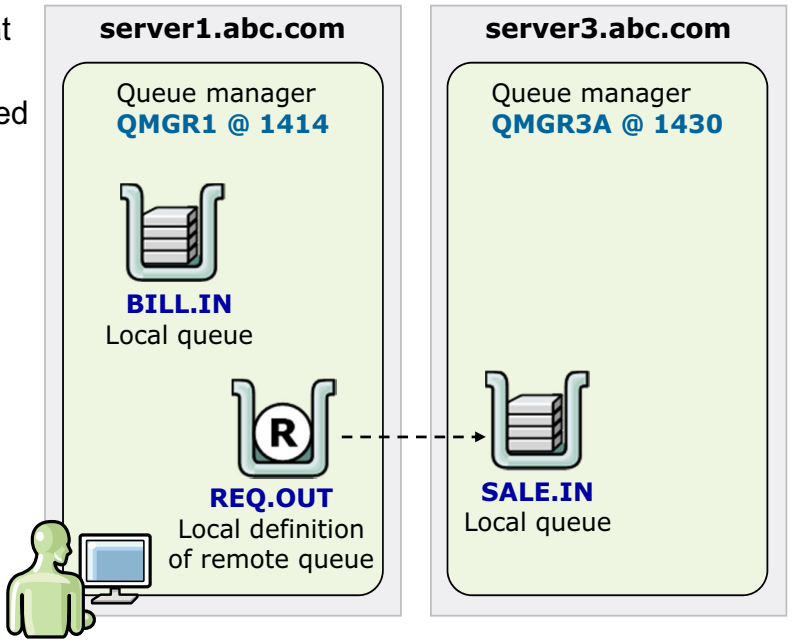


Figure 2-12. Terminology checkpoint

With the introduction of queues, it is important to distinguish between the local queues, remote queues, and local definitions of remote queues.

As noted in the queue manager terminology checkpoint, the terms *local* and *remote* are relative to the queue manager where work is done.

In the example, the administrator is connected to the server that is named `server1.abc.com`. So the local queue manager is QMGR1. The queue that is named BILL.IN is defined as a local queue on the queue manager. The queue manager also contains another queue that is named REQ.OUT. This queue is defined as a pointer to the remote queue SALE.IN on the remote queue manager that is named QMGR3A. This type of queue is also known as a *local definition of a remote queue*.

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

Figure 2-13. 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. Channels are used in distributed queuing to move messages from one queue manager to another.

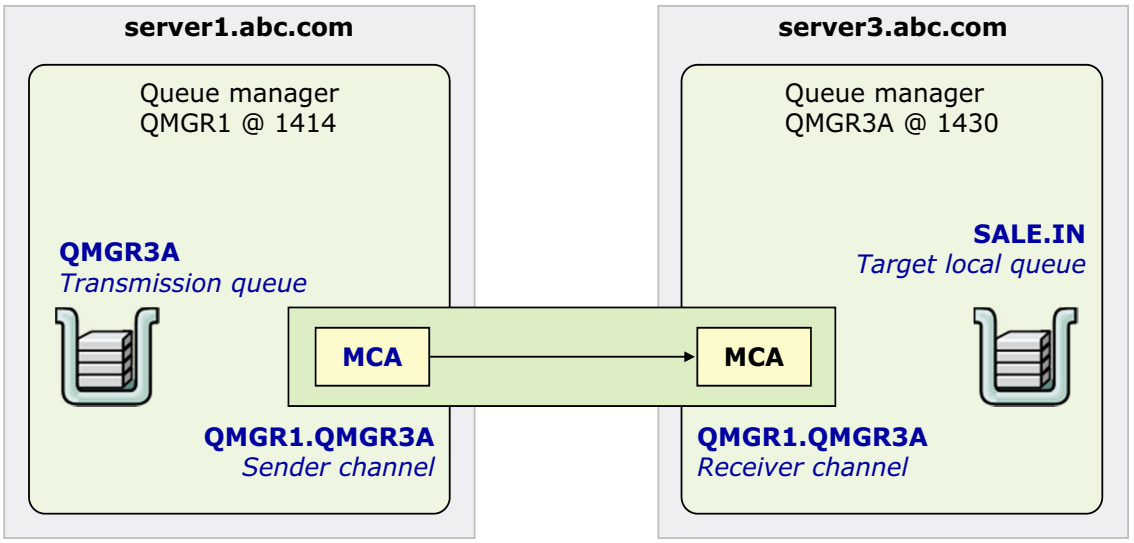
A *client channel*, also referred to as an *MQI channel*, provides a two-way communications link between a client application and a queue manager.

When you define message channels, you must define one channel object at the queue manager that is to send messages, and another, complementary one, at the queue manager that is to receive them. 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.

IBM Training

Message channel pair example: Sender-receiver



- Sender channel is a message channel that the queue manager uses to send messages to other queue managers
- To send messages by using a sender channel, you must also create, on the other queue manager, a receiver channel with the same name as the sender channel

Figure 2-14. Message channel pair example: Sender-receiver

This figure shows an example of how the message channels are paired to provide a communications link between two queue managers.

In the example, the queue manager that is named QMGR1 is configured with a *sender channel* that is named QMGR1.QMGR3A that can send message to the queue manager that is named QMGR3A. The queue manager that is named QMGR3A is configured with a *receiver channel* with the same name that can receive messages from the queue manager that is named QMGR3A.

During processing, the sender channel takes messages from transmission queue QMGR3A and sends them to the remote queue manager of the same name. It is a good practice to name the channels to be the same as the ‘from’ and ‘to’ queue managers, in the correct from-to order. If another channel pair had to be defined having the sender in QMGR3A and receiver in QMGR1, then each sender and receiver would be named QMGR3A.QMGR1.

IBM Training

Message expiration

- Expiry is a message property set by an MQ application that limits the time that a message can remain in a queue before it is processed
- Setting expiry is a good option for time-sensitive applications where messages become invalid if it remains in the queue longer than expected
- A special report message can be generated when a message expires

Figure 2-15. Message expiration

Expiry is a queue attribute and also an MQI option that identifies the time that a message remains valid after it is placed in a queue. Expiry is used in time-sensitive applications where the payload can become obsolete.

An expired message is normally discarded when an application attempts to obtain it. When a message with a set **Expiry** is placed on a queue, any stops on the message path towards its target further decrease the expiry time.

When a message expires, it can generate a report message.

Attention

Queue attributes and behavior might vary depending on the installation operating system. Always check the IBM Knowledge Center for IBM MQ Version 9.0 to identify any exceptions.

IBM Training

Report messages

- Messages that inform an application about events that pertain to the original message
  Examples:
  - Confirmation of arrival
  - Confirmation of delivery
  - Expiry report
- Can be configured to contain:
  - The entire original message
  - The first 100 bytes of the original message
  - Just the report and MQMD but none of the original message
- Using report messages must be part of the application design

Figure 2-16. Report messages

Report messages inform applications about events such as the occurrence of an error when a queue manager processes a message.

Report messages can be generated by an MQ queue manager, an MQ MCA, or an application.

When you put a message on a queue, you can select to receive different types of report messages.

- A *confirmation of arrival* report message indicates that the message reached its target queue.
- A *confirmation of delivery* report message indicates that the message was retrieved by a receiving application.
- An *expiry report message* indicates that an application attempted to retrieve a message that reached its expiry threshold; the message is marked to be discarded.

The report message can be configured to contain the entire original message, the first 100 bytes of data in the original message, or no data from the original message.

You can configure your application to use the report messages for troubleshooting, for example.

IBM Training

Application programming interfaces

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

Figure 2-17. 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 C, Visual Basic (on Windows), COBOL, Assembler (on z/OS), RPG (on IBM i), and PL/I (on z/OS). 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

MQI overview

- Consists of the following components:
  - *Calls* through which programs can access the queue manager and its facilities
  - *Structures* that programs use to pass data to, and get data from, the queue manager
  - *Elementary data types* for passing data to, and getting data from, the queue manager
- MQ sample programs demonstrate how to use the MQI

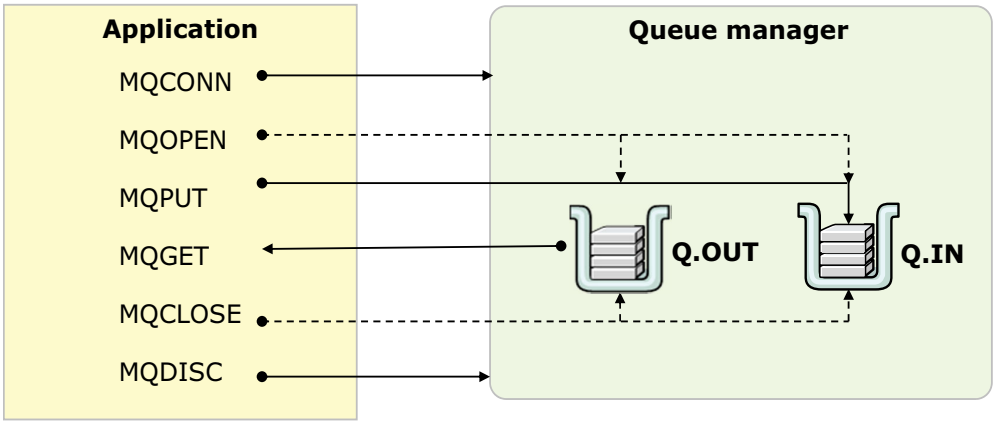


Figure 2-18. MQI overview

How do messages get placed into queues? The MQI provides access to the MQ services.

The MQI consists calls, structures, and elementary data types.

For example, applications connect to a queue manager and open a queue by using the MQI MQCONN and MQOPEN calls. If the call are successful, then messages can be put to queues by using the MQPUT call or gotten from a queue by using an MQGET call.

At the end of the process, the MQCLOSE call closes the queue and the MQDISC call disconnects the application from the queue manager. It is important to always close the queues and disconnect from the queue manager when finished placing messages in the queue.

MQ contains numerous code samples and compiled sample applications that can be used to test putting and getting messages from a queue.

IBM Training

Starting IBM MQ applications by using triggers

- IBM MQ can automatically start a process automatically when messages are available to retrieve
- Example processes:
  - Application program
  - Script
  - CICS transaction
  - Distributed message channel
- Triggering is engaged when the defined trigger conditions are met
- Supported by IBM MQ clients that are running on UNIX, Linux, and Windows

Figure 2-19. Starting IBM MQ applications by using triggers

Some MQ applications that serve queues can run continuously; the applications are always available to retrieve messages that arrive on the queues. However, you might not want this behavior when the number of messages that arrive on the queues is unpredictable. In this case, applications might be using system resources even when no messages are available to retrieve.

MQ provides a facility that enables an application to be started automatically when no messages are available to retrieve. This facility is known as *triggering*.

Triggering is an MQ capability to automatically start a process, such as an application program, that is based on attributes that the local queue defines.

Triggering provides automation to message consumers. For instance, it might be that all messages that arrive at the BILL.IN queue in queue manager QMGR3A need to be processed right away. How does the application that uses these messages know that messages are waiting to be processed in the queue? Triggering can be defined to serve this purpose.

IBM Training

Triggering: Process scenario

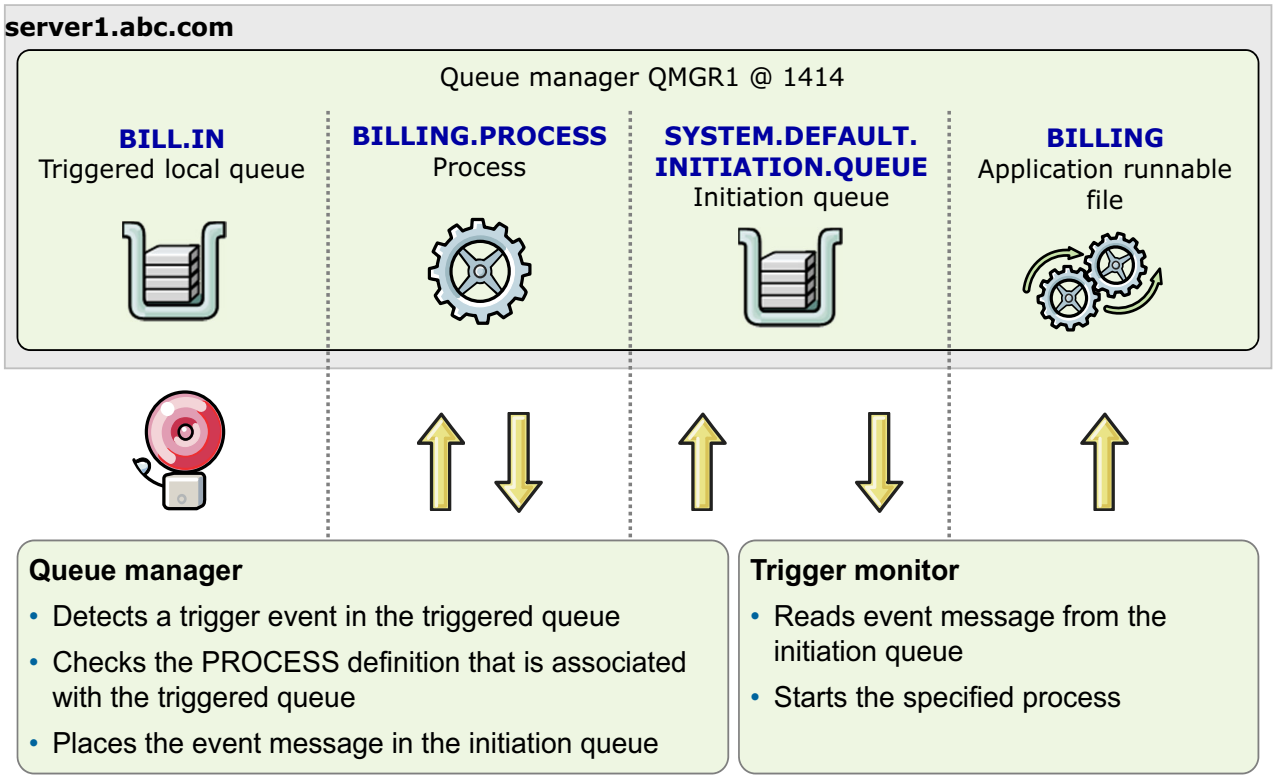


Figure 2-20. Triggering: Process scenario

To trigger a queue to start a process to get messages, you must define a process. The process definition contains details on where to find the program or script to start.

A local queue is the object that is triggered. Triggering attributes are set on the queue to indicate that triggering is to occur, and to specify the trigger conditions.

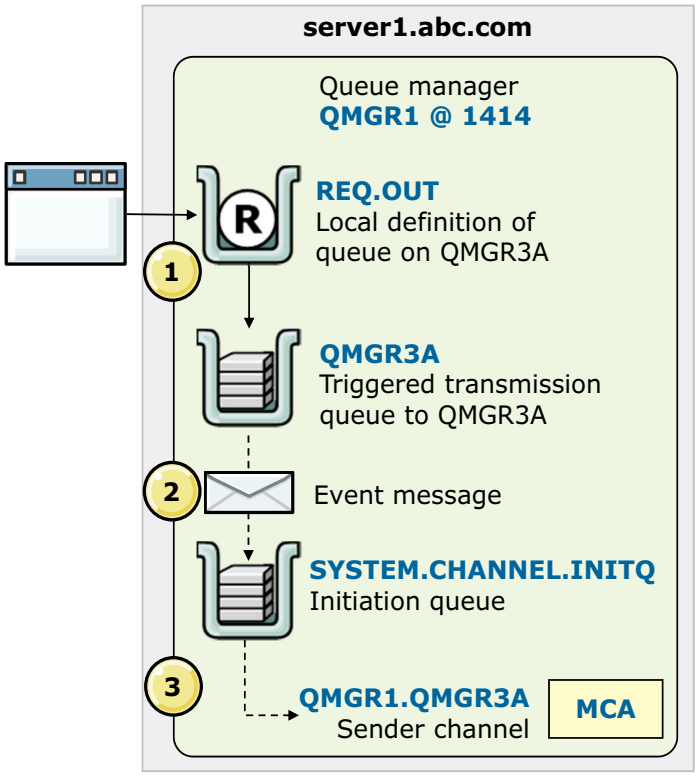
Another trigger components is the initiation queue. The initiation queue is another queue where the queue manager puts specially formatted trigger messages.

The two other triggering components are the queue manager and a process that is called the *trigger monitor*, which must be running for triggering to occur. You can also configure the trigger monitor to start when the queue manager starts.

Triggering required coordination between the queue manager and the trigger monitor. When the defined triggering conditions are met, the queue manager checks the process definition, then places a specially formatted message in the initiation queue. The initiation queue is under the watch of the trigger monitor, which gets the message from the initiation queue and starts the appropriate process.

IBM Training

Triggering: Channel scenario



1. The application puts a message to REQ.OUT, which causes the message to be placed in the QMGR3A triggered transmission queue.
2. The queue manager detects the message in the triggered queue and if trigger conditions are met, puts the event message in the initiation queue.
3. The *channel initiator* interprets the event message from the initiation queue and starts the sender channel

Figure 2-21. Triggering: Channel scenario

Setting up channels to start from a trigger is a special implementation of triggering. When using triggering to start channels, it is the transmission queue that gets triggered, not a process.

A special channel trigger monitor calls a channel initiator. When a message arrives in a triggered transmission queue for a channel that is not running, the queue manager places a special message in the SYSTEM.CHANNEL.INITQ initiation queue and the channel initiator starts the channel.

IBM Training

Transactions: Terminology

- *Resource manager* is a system that owns and controls its components such as
  - A database manager owns its tables
  - A queue manager owns its queues
- *Transaction* or *unit of work* is a set of changes that must be completed in their entirety (committed) or restored to a previous consistent state (backed out)
- *Transaction manager* is a subsystem that coordinates units of work
- MQ queue manager:
  - Acts as a transaction manager over its own resources
  - Manages updates to DB2 tables
  - Runs under a compatible external transaction manager

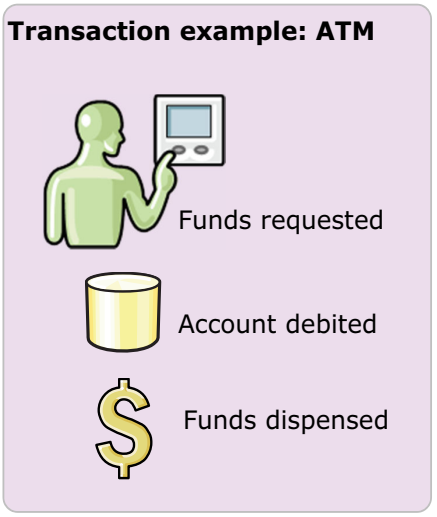


Figure 2-22. Transactions: Terminology

This slide summarizes terminology that is associated with a transaction.

To review, a *resource manager* is a system, such as a queue manager, that owns and controls its components.

A *transaction* is a set of changes that must be completed or restored in their entirety. An ATM withdrawal is an example of a transaction.

A *transaction manager* is a subsystem that coordinates units of work. MQ can serve as a transaction manager, or it can defer resource management to a compatible transaction manager such as CICS.

IBM Training

Transactions: Global units of work

- Transactions that must be coordinated across two or more resource managers by using a *two-phase commit* process
- Might involve different queue managers and database tables

Using an IBM MQ queue manager as the transaction manager

- Supported on Linux, UNIX, and Windows only
- Unit of work is started with MQBEGIN MQI function
- Unit of work is completed with MQCMIT or MQBACK MQI function

```
MQCONNECT
MQOPEN
MQBEGIN
MQPUT
SQL INSERT
MQCMIT or MQBACK
MQCLOSE
MQDISC
```

Using an external transaction manager

- External transaction manager must be compatible (X/Open XA compliant)
- MQ queue manager is a participant but does not manage the transaction
- Application requests the external transaction manager (TM) such as CICS and Microsoft Transaction Server to start the unit of work
- Transaction is controlled with TM API
- Requirements for each TM must be confirmed

Figure 2-23. Transactions: Global units of work

Transactions can be coordinated across two more resource managers under global transaction management.

When you are implementing a solution that uses MQ, you can use MQ as the transaction manager or you can use an external transaction manager.

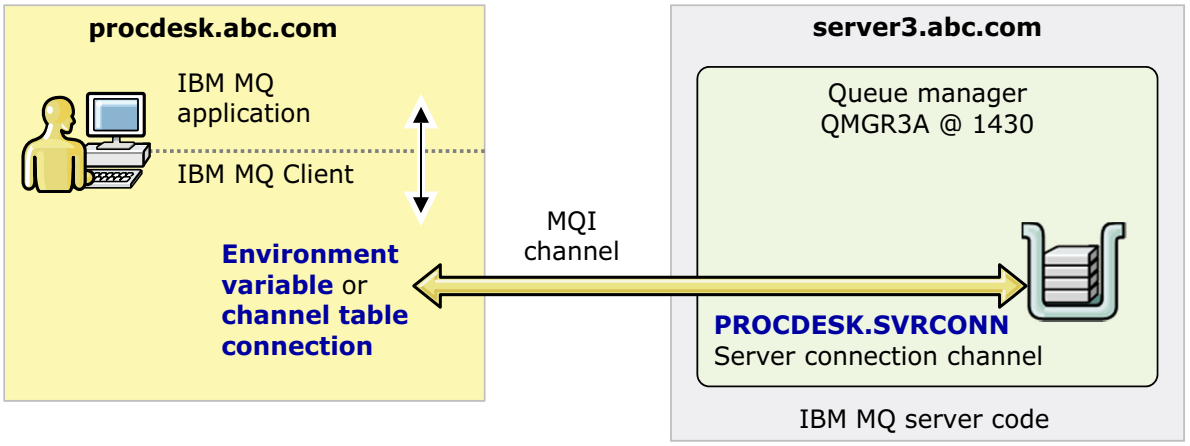
This slide summarizes the differences between using MQ or not using MQ as the transaction manager. For example, if the transaction is included in a global unit of work and if MQ is used as the resource coordinator, the MQ MQI functions are used to commit or back out the transaction. If an external transaction manager is used, then the transaction manager API is used to commit or roll back changes.

Attention

If you use an external transaction manager, you must ensure that it is compatible with IBM MQ.

IBM Training

IBM MQ client



- MQ runtime component that does not require queue manager runtime code
- Enables an application, running on a server where only the client runtime code is installed, to connect to a queue manager that is running on another server and complete messaging operations with that queue manager
- Uses the same MQI calls as MQ server
- Flow of messages is bidirectional
- MQ client has partial transactional capabilities

IBM MQ basics

Figure 2-24. IBM MQ client

An MQ clients is a compact version of MQ with many of the same functions. Unlike the MQ server installation where MQ must be configured and objects are defined after installation, an MQ client has a connection to a queue manager server-connection channel only.

An MQ client does not require a separate MQ license.

Applications for an MQ client use the same function calls as the MQI, but they are compiled with MQ client libraries instead of MQ server libraries. MQ clients can also participate in a transaction, but only when the queue manager to which the MQ client is connected controls this transaction.

IBM Training

Managing client connections

- On queue manager, define a server-connection channel and a client-connection channel
- On client workstation, make client client-connection definition available to the client by using the client-channel definition table (CCDT)
  - Option 1: Copy the CCDT to the client computer
  - Option 2: Copy the CCDT to a location shared by more than one client
  - Option 3: Leave the CCDT on the server but make it shareable by the client
- IBM MQ Version 9.0 improves the ability for clients to remain connected to queue managers by hosting the CCDT in a central location that is accessible through a URL

Figure 2-25. Managing client connections

As noted previously, an MQ client uses a client MQI channel that can send data in both directions.

To connect the MQ client to a queue manager, you must define both a server-connection channel and a client-connection channel.

You can create both definitions on the server, then make the client-connection definition available to the client. On all supported operating systems, you can use MQ Script (MQSC) commands, programmable command format (PCF) commands or the IBM MQ Explorer to define a server-connection channel on the server computer.

The client-connection channel definitions that are created on the server are made available to clients by using a client channel definition table (CCDT). The CCDT determines the channel definitions and authentication information that is used by client applications to connect to the queue manager. On operating systems other than z/OS, a CCDT is created automatically. You must then make it available to the client application by copying it to the client computer or to a shared location.

In IBM MQ Version 9.0, you can put the CCDT in a shared location that is accessible by using a URL. This enhancement removes the need to individually update the CCDT for each deployed client.

IBM Training

IBM MQ clients

IBM MQ client

- Cannot update resources that another resource manager owns
- Can participate in a unit of work that is managed by the queue manager to which it is connected
- Can use MQI MQCMIT or MQBACK functions to complete the unit of work

IBM MQ extended transactional client

- Can put to and get messages from the queue manager to which it is connected
- Can update resources that another resource manager in the same unit of work owns
- An external transaction manager collocated with the client application must manage unit of work
- Uses external transaction manager API
- Not available on z/OS but a client application that uses an extended transactional client can connect to a z/OS queue manager

Figure 2-26. IBM MQ clients

A client application can participate in a unit of work that is managed by a queue manager to which it is connected. Within the unit of work, the client application can put messages to, and get messages from, the queues that are owned by that queue manager. The client application can then use the MQI MQCMIT call to commit the unit of work or the MQI MQBACK call to back out the unit of work. However, within the same unit of work, the client application cannot update the resources of another resource manager, such as the tables of a DB2 database. An MQ extended transactional client removes this restriction.

An MQ extended transactional client is an MQ MQI client with some additional features. An MQ extended transactional client can do the following tasks within the same unit of work:

- Put messages to, and get messages from, queues that are owned by the queue manager to which it is connected.
- Update the resources of a resource manager other than an MQ queue manager.

Extended transactional clients are not available for z/OS. A client application that is using an extended transactional client can connect to a queue manager that runs on z/OS.

IBM Training

Unit summary

- Describe the components of IBM MQ
- Describe the role of a channel in distributed queuing
- Describe how a message gets from source to target queue and where messages are at different times in the process
- Summarize process triggering
- Summarize options for transaction handling
- Differentiate between an IBM MQ server and an IBM MQ client
- Describe the IBM MQ extended transactional client

Figure 2-27. Unit summary

Review questions

1. Which of the following queues can hold messages?
   A. A remote queue
   B. The dead-letter queue
   C. A local queue
   D. A transmission queue
2. True or False: Client channels require a transmission queue to send messages to a queue manager.

Figure 2-28. Review questions

Write your answers here:

1.

2.

IBM Training

Review answers

1. Which of the following queues can hold messages?
   A. A remote queue
   B. The dead-letter queue
   C. A local queue
   D. A transmission queue

   The answer is B, C, and D.

2. True or False: Client channels require a transmission queue to send messages to a queue manager.

   The answer is False. Distributed channels that send messages between queue managers require a transmission queue on the sender queue manager; client channels do not require a transmission queue.

Figure 2-29. Review answers

Unit 3. Messaging styles, topologies, and architecture overview

Estimated time

00:30

Overview

This unit expands on IBM MQ connectivity and architectural topics and describes scalability and high availability options.

How you will check your progress

- Review questions

References

IBM MQ Library: http://www.ibm.com/software/integration/wmq/library

IBM Training

Unit objectives

- Differentiate between point-to-point and publish/subscribe messaging styles
- Describe the advantages of using queue manager clusters
- Identify the options for high availability, scalability, and load balancing
- Describe the need for standards and governance

Figure 3-1. Unit objectives

This unit describes publish/subscribe messaging style and compares it to the point-to-point style described in previous units. This unit also introduces MQ clustering and at how to provision a highly available MQ infrastructure.

A robust infrastructure is the responsibility of everyone in the organization, which includes architects, developers, business users, and planners, not just the MQ administrator. The end of this unit describes some lessons that were learned from actual MQ engagements, which highlight the criticality of standards and governance.

IBM Training

Point-to-point and publish/subscribe

- Point-to-point application
  - Simplest form of messaging in IBM MQ
  - Sending application must know certain information about the receiving application before messages can be sent
  - Sending application sends messages to a predefined queue
- Publish/subscribe application
  - Publishes messages to an interim destination according to a topic
  - Interested recipients subscribe to the topic
  - No explicit connection between publishing and subscribing applications

Figure 3-2. Point-to-point and publish/subscribe

A point-to-point application is the simplest form of messaging in MQ. In a point-to-point application, the sending application provides specific information about the receiving application.

In a publish/subscribe application, the publisher of information is further decoupled, and not directly connected, to the subscriber of that information.

- Publishers are providers of information. Publishers do not need to know the subscribers because connectivity is deferred to the infrastructure.
- Subscribers are consumers of information and do not need to know providers.
- New publishers and subscribers can be added to the network without disruption.

IBM Training

Point-to-point and publish/subscribe comparison

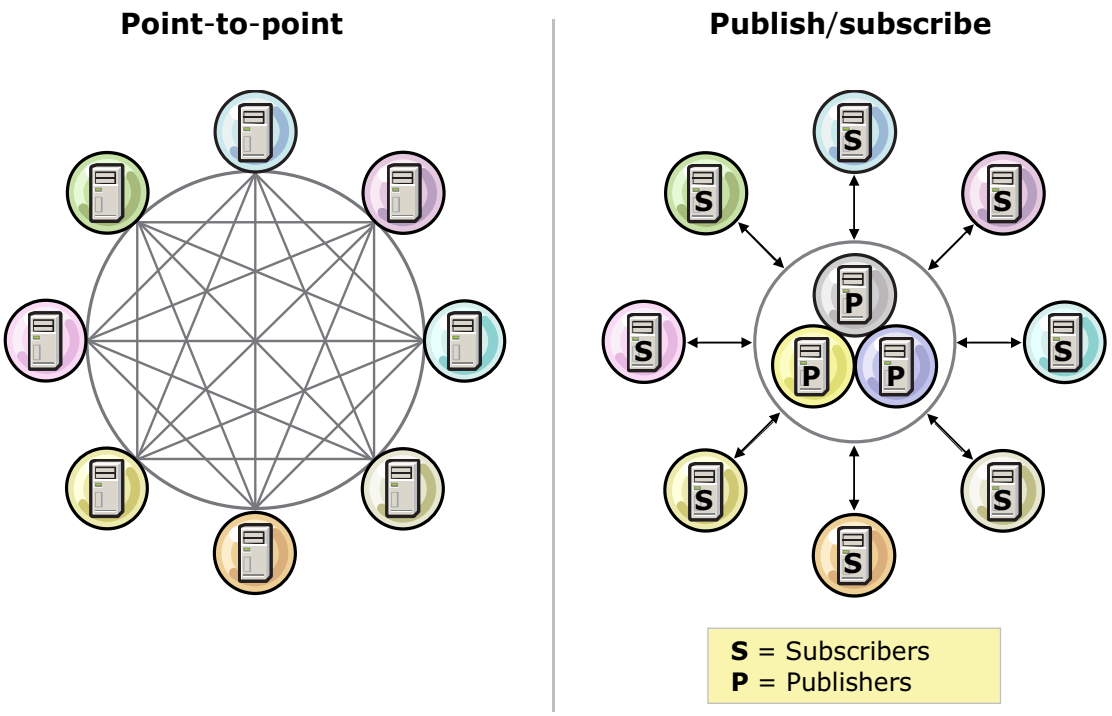


Figure 3-3. Point-to-point and publish/subscribe comparison

As shown on the left side of the figure, the point-to-point messaging style requires connections across all queue managers that need to communicate.

As shown on the right side of the figure, publish/subscribe needs a connection between the publisher and subscriber. After the publish/subscribe infrastructure is defined, no additional connectivity needs to be defined unless new servers or queue managers are added to the infrastructure.

The publish/subscribe engine handles routing of messages by using publications and subscriptions. The sender of a publication does not need to know where the messages are going. The receiver does not need to know where messages originated, only that they subscribed to the correct topic.

IBM Training

Point-to-point messaging example

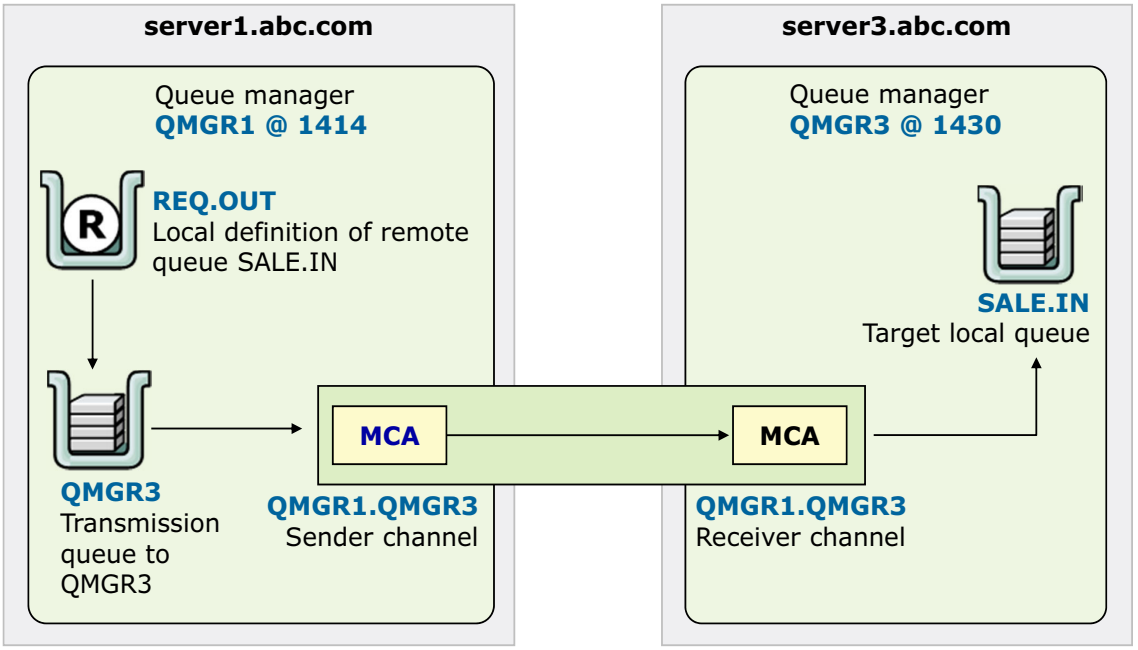


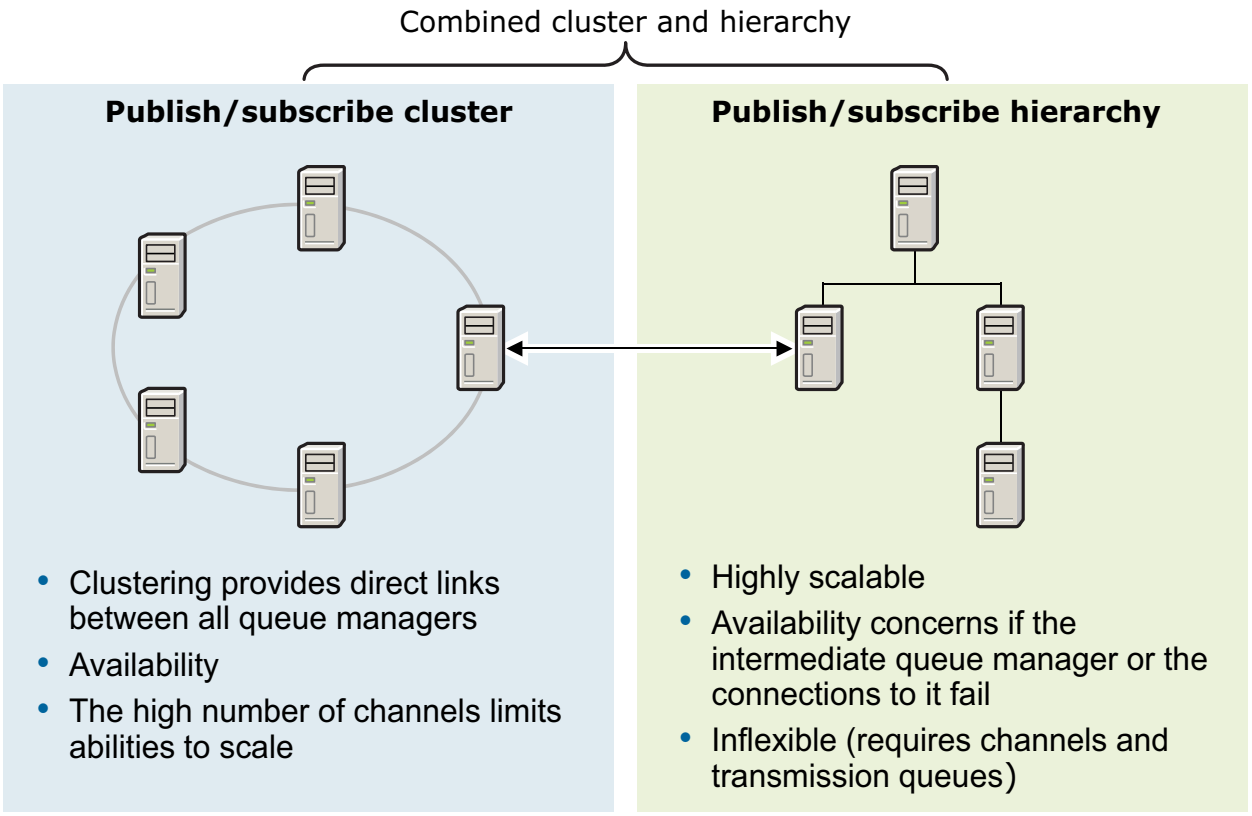
Figure 3-4. Point-to-point messaging example

This figure reviews the MQ components for point-to-point messaging.

In this design, a message that is placed in queue remote REQ.OUT in QMGR1 would eventually be available for consumption at the SALE.IN local queue in QMGR3. You do have a degree of decoupling because, if needed, you can change REQ.OUT to point to another queue manager and another queue without any impact to the application. However, you would still know where the message is going.

IBM Training

Distributed publish/subscribe topologies



Messaging styles, topologies, and architecture overview

Figure 3-5. Distributed publish/subscribe topologies

Queue managers that are connected together into a distributed publish/subscribe topology share a common topic space. Subscriptions that are created on one queue manager can receive messages that are published by an application that is connected to another queue manager in the topology.

You can control the extent of topic spaces that are created by connecting queue managers together in clusters or hierarchies.

When you design a publish/subscribe infrastructure, it is critical to focus on the traits of the infrastructure that are optimal for the organization. Is a cluster architecture best? Would a hierarchy be more appropriate? Or is a combined infrastructure optimal?

A cluster provides direct links between all queue managers. In a publish/subscribe cluster, a topic object must be 'clustered' for each branch of the topic space that is to span the cluster.

In a hierarchy, you connect a child queue manager to a parent queue manager in the hierarchy. If the child queue manager is already a member of another hierarchy or cluster, then this connection joins the hierarchies together, or joins the cluster to the hierarchy.

IBM Training

What is a queue manager cluster?

- Group of queue managers that make the queues that they host available to other queue managers in the cluster without explicit channel definitions, remote-queue definitions, or transmission queues for each destination

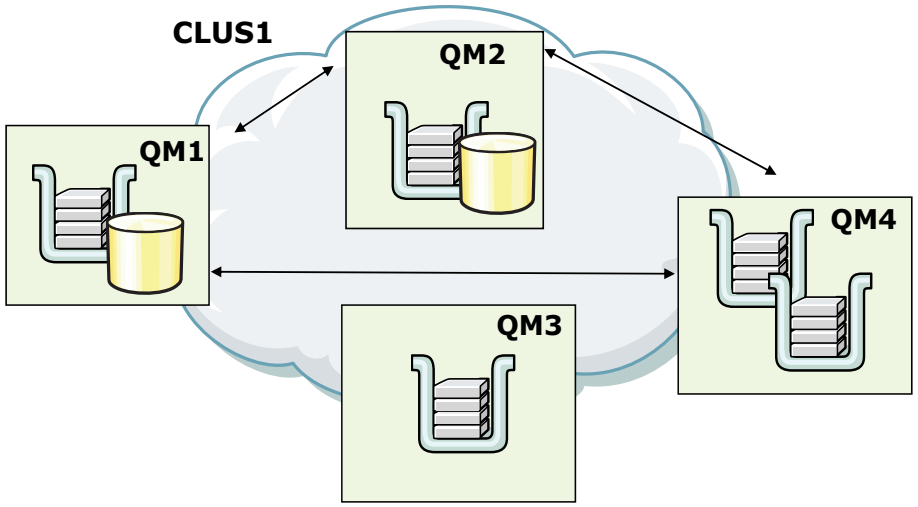


Figure 3-6. What is a queue manager cluster?

A 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.

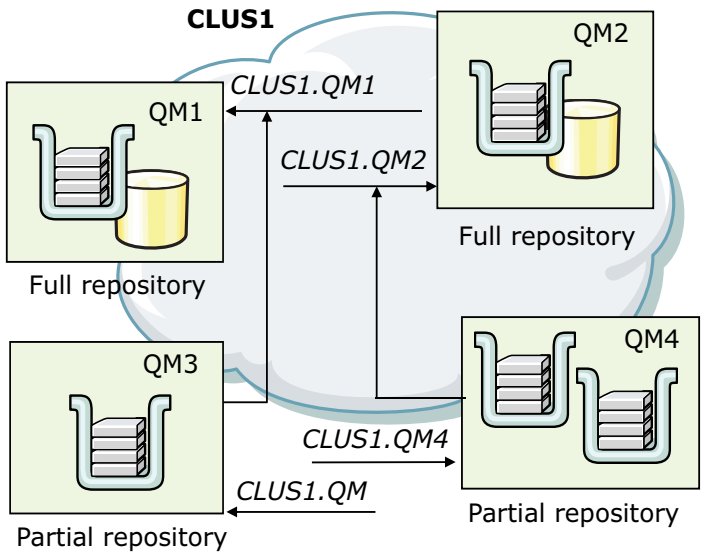
Cluster-specific objects remove the need for explicit channel definitions and transmission queues for each destination queue manager.

The queue managers in a cluster often assume the role of a client or a server. The servers host the queues that are available to the members of the cluster and applications that process these messages and generate responses. The clients PUT messages to the server queues and might receive response messages.

Queue managers in a cluster normally communicate directly with each other, although typically, many of the client systems never need to communicate with other client systems.

IBM Training

Queue manager cluster components



- *Full repository queue managers* host a complete set of information about every queue manager in the cluster
- *Partial repository queue managers* contain information about those queue managers with which it needs to exchange messages
- *Cluster queues*
  - Application queues
  - Transmission queues
- *Cluster channels*
  - Cluster-receiver
  - Cluster-sender

Figure 3-7. Queue manager cluster components

With MQ, information about clustered queues and connectivity is kept in a *cluster repository*. The cluster is set up by identifying two queue managers to hold a full cluster repository and by defining cluster channels across all queue managers that are in the cluster. After these definitions are complete, the cluster process “learns” how to find any queues that are part of the cluster.

The other queue managers in the cluster are *partial repository queue managers*. These queue managers contain information about the queue managers with which it must exchange messages.

All the queue managers in the cluster can host local queues that are accessible to any other queue manager in the cluster. These queues are called *cluster queue*s.

The *cluster channels* exchange cluster information with one of the full cluster repositories.

By default, cluster queue managers use a system queue that is called SYSTEM.CLUSTER.TRANSMIT.QUEUE to send messages. It is also possible to designate a separate transmit queue to offer process isolation.

The example in the figure contains four queue managers in the cluster that is named CLUS1. Queue managers QM1 and QM2 are full repository queue managers. The repositories are represented in the figure by the shaded cylinders.

IBM Training

Clustered queues and channels

- Clustered queues
  - Application cluster queues that a cluster queue manager hosts and makes available to other queue managers in the cluster
  - Cluster transmission queues hold messages for clustered queue managers
- Clustered channels
  - Cluster-receiver channel defines the end of a channel on which a cluster queue manager can receive messages from other queue managers in the cluster
  - Cluster-sender channel communicates any cluster-related changes to the full repository queue manager

Figure 3-8. Clustered queues and channels

As with distributed queuing, an application can put a message on a cluster queue at any queue manager. An application can get a message from a cluster queue on the queue manager to which it is connected.

When a queue manager joins the cluster, it uses its cluster-sender channel to send information about itself to its designated full cluster repository queue manager. If a clustered queue is defined in a queue manager, that queue manager uses its cluster-sender channel to send information about the new queue to its full cluster repository.

When an application needs to send a message, MQ uses the information in the cluster repository to create a temporary dynamic sender channel to the intended queue manager's cluster-receiver channel.

IBM Training

Benefits of IBM MQ clusters

- Simplified administration
  - Reduced number of remote queues, transmission queues, and channel definitions
- Workload balancing
  - Same queue can be hosted in several queue managers
  - Route around failures
- Scalable
- Publish/subscribe can use a cluster

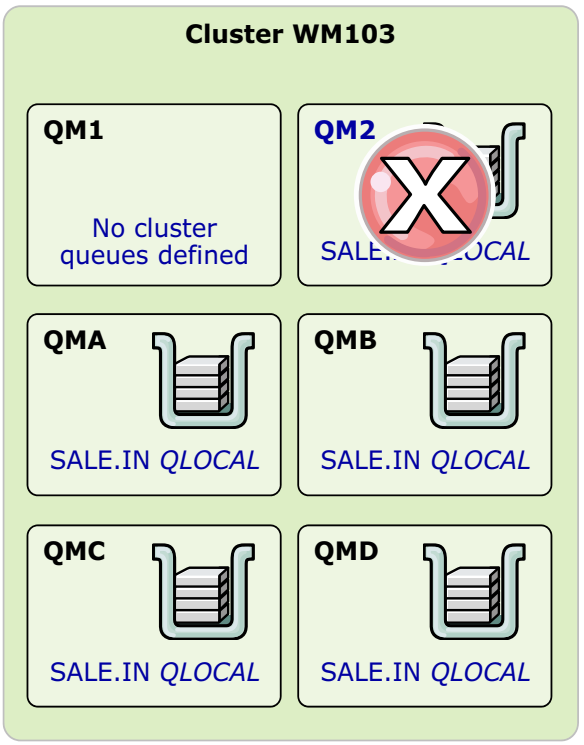


Figure 3-9. Benefits of IBM MQ clusters

Organizations have different reasons for using MQ clusters.

One benefit of MQ cluster is simplified administration. For example, if you had four queue managers that required interconnectivity and you did not use a cluster, you would need to define 24 objects. If the queue managers are in a cluster, you need to define eight objects only.

Another benefit is workload balancing. When queue managers join a cluster, they can offer to host a copy of the same queue so several servers can service the application. If one queue manager fails, another queue manager in the cluster can take over and continue providing service to the application.

IBM Training

Highly available IBM MQ topologies

- MQ clusters offer scalability by having multiple queue managers to service a request
  - Clustering provides availability for new incoming requests
  - When a queue manager fails, messages in transit for that queue manager are marooned
- If the queue manager is a critical resource such as a gateway, there must be a way to replace, or fail over the queue manager
  - Queue managers that are configured for failover must be on shared disk so messages from a failed queue manager are accessible by the spare queue manager
  - Shared disk must be mirrored synchronized for integrity
- MQ supports two failover configurations
  - High-availability clusters with IBM PowerHA for AIX (formerly HACMP), Veritas Cluster Server, Microsoft Cluster Server, HP Serviceguard
  - Multi-instance queue managers

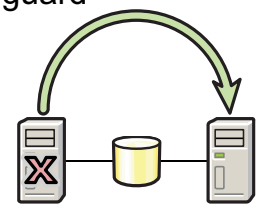


Figure 3-10. Highly available IBM MQ topologies

A concern for enterprise messaging is high availability.

An MQ cluster by itself is not the solution to all failover problems. An MQ cluster helps in servicing new requests, but what happens to messages that were in-process on the queue manager that failed? These messages, normally referred to as “marooned messages,” are inaccessible as they are in the failed queue manager. To be able to pick up those marooned messages, you need to implement some form of failover mechanism for the queue manager.

MQ queue managers can use external high availability software to provide failover capabilities. For operating systems other than z/OS, MQ provides its own failover implementation that uses *multi-instance queue managers*.

IBM Training

Multi-instance queue managers: All is well

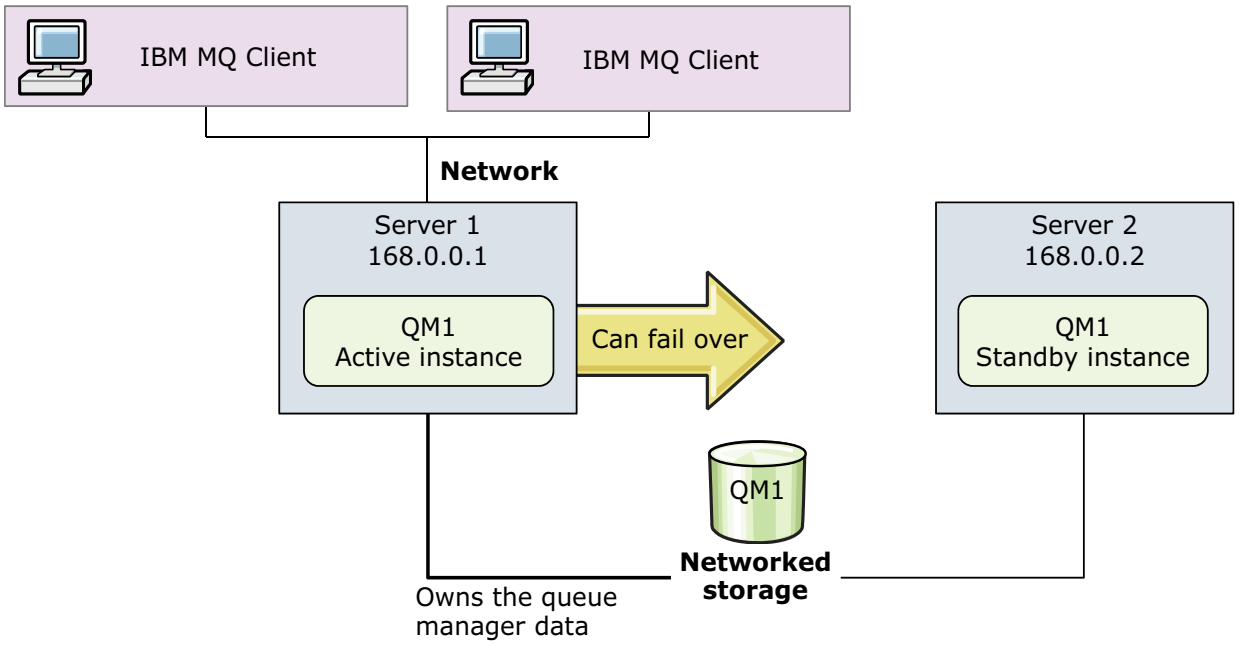


Figure 3-11. Multi-instance queue managers: All is well

Multi-instance queue managers provide a failover option for queue managers.

Multi-instance queue managers are instances of the same queue manager that are configured on different servers. One instance of the queue manager is defined as the active instance and another instance is defined as the standby instance. If the active instance fails, the multi-instance queue manager restarts automatically on the standby server.

The active instance has exclusive access to the shared queue manager data and logs folders when it is running.

A multi-instance queue manager is one part of a high availability solution. You must also provide a high-performance shared network file system that manages locks correctly and provides protection against media and file server failure.

IBM Training

Multi-instance queue managers: Problem occurs

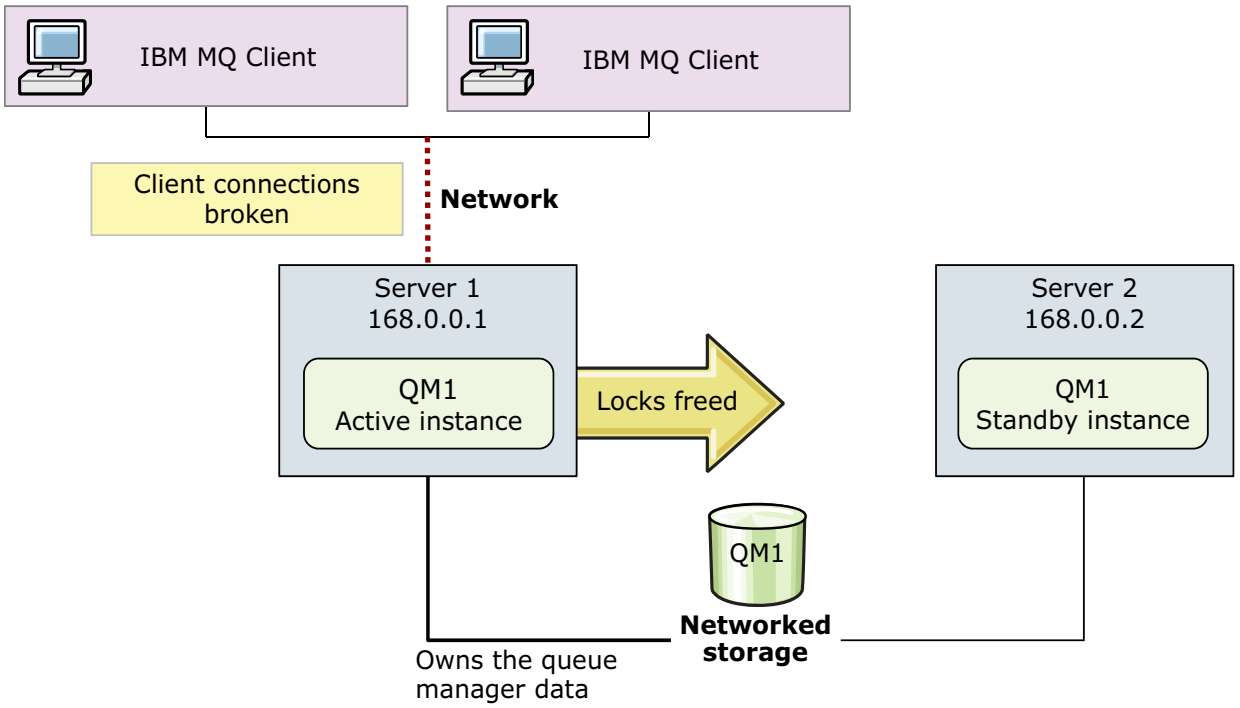


Figure 3-12. Multi-instance queue managers: Problem occurs

When QM1 fails, client connections are broken and the locks on the networked storage are freed.

IBM Training

Multi-instance queue managers: Fail over, then reconnect

1. Queue manager fails over to the standby queue manager
2. Shortly after the failover occurs, the connections are restored

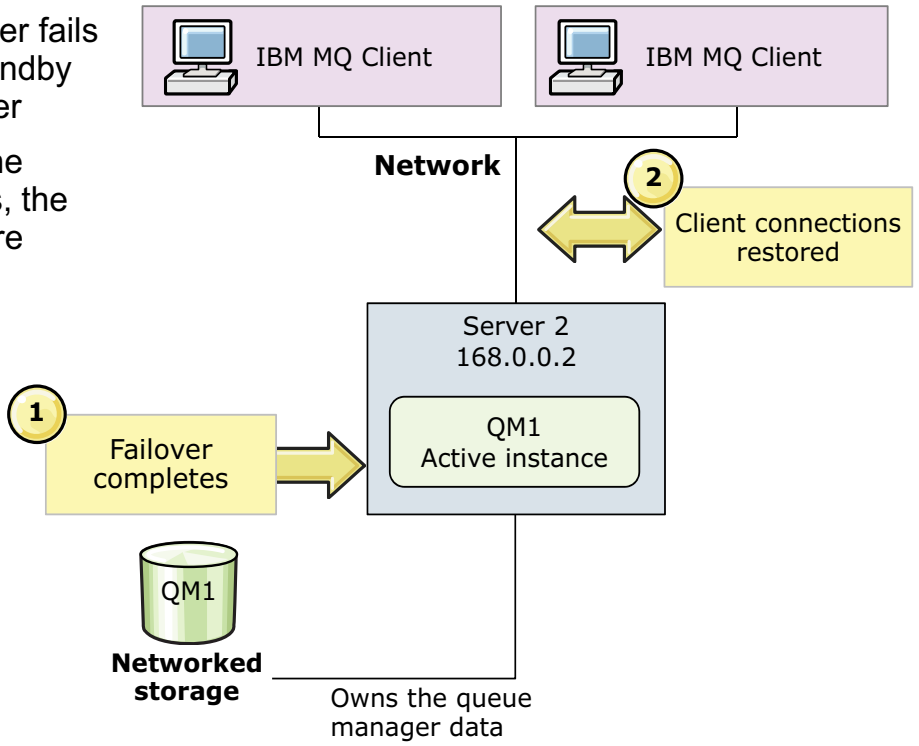


Figure 3-13. Multi-instance queue managers: Fail over, then reconnect

When the standby instance of QM1 detects that the active instance failed, it becomes the active instance. It takes over the QM1 data and logs in the state they were left by the active instance, and accepts reconnections from clients and channels.

Multi-instance queue managers fail over queue managers only. They are not a complete high availability solution because they do not handle other applications in the server. If the server is dedicated to MQ and no other software components need to be failed over, multi-instance queue managers are a failover option.

IBM MQ failover technology comparison

Multi-instance queue manager

Advantages

- Integrated into the IBM MQ
- Faster failover than HA cluster
  - Shorter delay before the queue manager restarts

Limitations

- Runtime performance of networked storage
- More complex network configuration because queue manager changes IP address when it fails over

HA cluster

Advantages

- Can coordinate multiple resources
- More flexible configuration options
- Can fail over multiple times without operator intervention
- Takeover of queue manager's IP address as part of the failover

Limitations

- Unnecessary failovers cause frustration
- Extra product purchase and skills required

Figure 3-14. IBM MQ failover technology comparison

This figure summarizes the factors to consider when you choose a failover technology.

For example, one of the limitations of an HA cluster is the extra cost and administration skills. If your staff is already familiar with HA software, it might be better to use an HA cluster instead of a multi-instance queue manager.

IBM Training

Highly scalable IBM MQ topologies

Different technologies to address different needs	Fail over with shared disks	IBM MQ clusters
Immediate availability upon loss of an MQ queue manager (new requests versus waiting for failover to occur)	No	Yes
Processing stranded, or “marooned” messages (requests during failure)	Yes	No
Scalability and workload balancing	No	Yes

Figure 3-15. Highly scalable IBM MQ topologies

This figure summarizes the technologies available for a highly scalable MQ topology.

An MQ cluster alone provides some failover, scalability, and workload balancing but does not handle “marooned” messages.

Shared disks can help with “marooned” messages but cannot provide workload balancing and scalability.

A combination of MQ clusters and a failover solution, can provide a highly scalable solution because each component handles different situations.

IBM Training

Terminology checkpoint: Multi-instance and multi-version

- Multi-instance queue manager refers to the ability to use the IBM MQ technology for failover of the MQ queue manager
- Multi-version installation refers to the ability to install one or more versions of IBM MQ in the same server

MQ multi-version environment display

```
InstName:       Installation1
InstDesc:       MQ7502
Identifier:     1
InstPath:       C:\IBM\WebSphere MQ
Version:        7.5.0.2
Primary:        Yes
State:          Available
MSIProdCode:    {38E913AA-0F10-434C-BEEC-
7473D6C196E8}
MSIMedia:       7.5 Server
MSIInstanceId: 1

InstName:       MQV9install
InstDesc:
Identifier:     2
InstPath:       C:\mqv9
Version:        9.0.0.0
Primary:        No
State:          Available
MSIProdCode:    {74F6B169-7CE6-4EFB-8A03-
2AA7B2DBB57C}
MSIMedia:       9.0 Server
MSIInstanceId: 1
```


Figure 3-16. Terminology checkpoint: Multi-instance and multi-version

It is important to distinguish between a multi-instance queue manager and a multi-version installation.

The practice of having more than one version of MQ installed is called a “multi-version” installation. A multi-version installation might have WebSphere MQ V8 and MQ V9 installed in the same server.

Multi-version installations do not provide failover capabilities. It means that the same server can be used for more than one version of MQ. The terms are similar and can be confusing.

IBM Training

The case for standards and governance

An MQ infrastructure can grow rapidly.

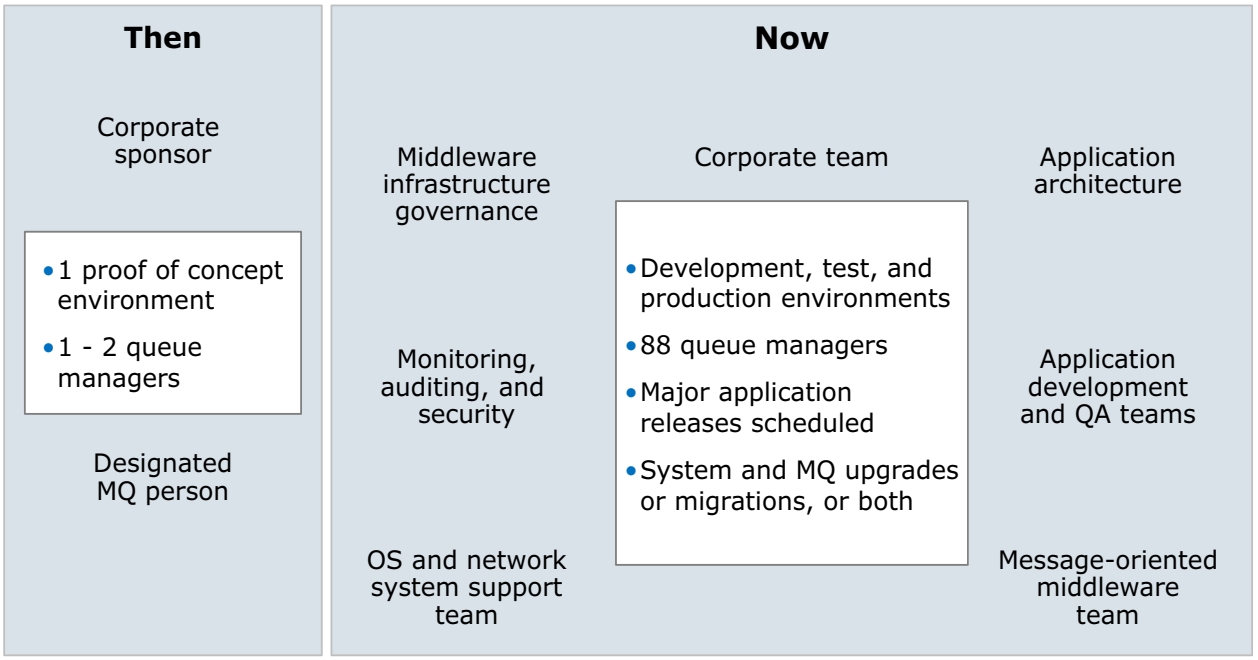


Figure 3-17. The case for standards and governance

When MQ is introduced in an organization, it can quickly expand into parts of the enterprise.

A proof-of-concept environment with 1 or 2 queue managers can turn into a complex infrastructure with three or more environments and many queue managers.

Regardless of role, individuals in organizations have different priorities.

- The corporate sponsor wants to show results.
- Developers need to define application queues.
- Testers need to get the partner application's queue manager connected.
- Architects must design sound, robust systems; but without a basic understanding of MQ they might introduce problems.

Often, everyone in the organization contacts the MQ administrator for a solution. However, the MQ administrator should not have full responsibility for the overall outcome of bad design and configuration requests that an administrative gatekeeper should review.

Appropriate MQ guidelines are necessary.

- Standardize naming conventions
- Establish administrative and operational standards
- Develop architectural guidelines that include distributed, clustering, and publish/subscribe
- Determine security requirements and standards

IBM Training

Unit summary

- Differentiate between point-to-point and publish/subscribe messaging styles
- Describe the advantages of using queue manager clusters
- Identify the options for high availability, scalability, and load balancing
- Describe the need for standards and governance

Figure 3-18. Unit summary

IBM Training

Review questions

1. Which statements are true for publish/subscribe?
   A. Messages are published to a topic.
   B. Applications that are interested in the topic subscribe to receive the messages.
   C. A direct connection must exist between publisher and subscriber.
   D. Publish/subscribe has evolved over time.

2. What statements are correct considerations when designing a highly scalable MQ infrastructure?
   A. Use an MQ cluster environment to promote scalability.
   B. Use multi-version queue manager installations to provide failover.
   C. Use a form of failover with shared disks to handle marooned messages.
   D. Define a clustered queue in more than one cluster member queue manager to ensure that any new request continues to be processed even after one queue manager fails.

Figure 3-19. Review questions

Write your answers here:

1.

2.

IBM Training

Review answers

1. Which statements are true for publish/subscribe?
   A. Messages are published to a topic.
   B. Applications that are interested in the topic subscribe to receive the messages.
   C. A direct connection must exist between publisher and subscriber.
   D. Publish/subscribe has evolved over time.

   The answer is A, B, and D.

2. What considerations are correct when designing a highly scalable MQ infrastructure?
   A. Use an MQ cluster environment to promote scalability.
   B. Use multi-version queue manager installations to provide failover.
   C. Use a form of failover that is coupled with shared disks to handle marooned messages.
   D. Define a clustered queue in more than one cluster member queue manager to ensure that any new requests continue to be processed even after one queue manager fails.

   The answer is A, C, and D.

Figure 3-20. Review answers

Unit 4. System administration overview

Estimated time

00:45

Overview

This unit identifies the administrative interfaces for IBM MQ, summarizes the basic administrative tasks, and examines use of the IBM MQ Explorer.

How you will check your progress

- Review questions

References

IBM MQ Library: www.ibm.com/software/integration/wmq/library

IBM Training

Unit objectives

- Summarize the system administration interfaces for IBM MQ
- Identify basic differences between IBM MQ on distributed operating systems and z/OS
- Contrast administrative interfaces according to role
- Summarize the system administration tasks
- Describe the installation options
- Describe the concepts of logging and recovery
- Summarize the administrative features of IBM MQ Explorer

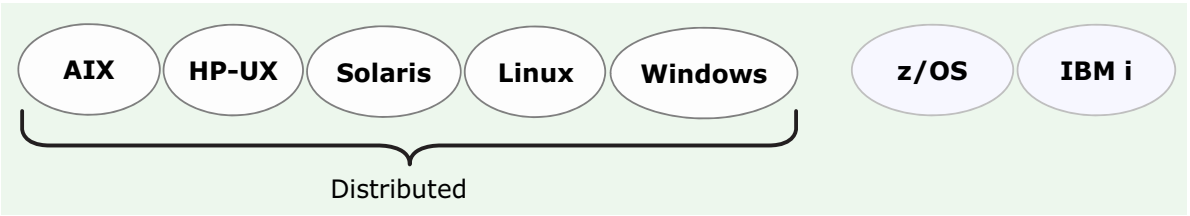
Figure 4-1. Unit objectives

IBM Training

Choosing an IBM MQ administration method

The administration method to use is determined by factors such as:

- The task
- The operating system



- The location of the MQ (local or remote)
- The user's role and permissions
- Personal preference

Figure 4-2. Choosing an IBM MQ administration method

In an organization, several people access MQ in different ways. While administrators work with configuration, business users need to access MQ for testing purposes.

MQ provides many tools and options for administrative access to its components. The method to access MQ should be the one that is best suited to the needs of the user, the MQ operating system, and the location of the MQ server.

IBM Training

Different administrative needs

Administrative users

- Responsible for the overall welfare of the queue manager
- Need to log on to MQ server
- Need an administrative ID with MQ privileges
- Need batch type functions to define and back up the environment

Developers, testers, and business users

- Responsible for tasks that are related to testing and quality assurance
  - Did the message arrive?
  - Is the channel running?
  - How many messages are in the queue?
  - Is the data formatted correctly?
- Logging on to MQ server is not required

Figure 4-3. Different administrative needs

MQ users are typically divided into two categories: administrative users and developers, testers, and business users.

Administrative users are responsible for the operation of the queue managers, MQ services, and components such as queues and channels. The authorizations that administrators need include the ability to create and delete queues. Administrators also need batch type functions so that they can complete efficiently reproduce administration tasks.

The needs of developers, testers, and business users include the ability to connect to a queue manager, open a queue, and view a message.

IBM Training

IBM MQ administrative tasks

- Install, apply maintenance, and configure:
  - IBM MQ
  - IBM MQ Managed File Transfer
  - IBM MQ Advanced Message Security
  - IBM MQ Telemetry
- Create queue managers
- Create queues, clusters, channels, or any objects that users request
- Administer publish/subscribe topologies
- Administer security and SSL keystores and truststores
- Back up file systems and object definitions
- Monitor disk space
- Identify problems
- Participate in infrastructure and capacity planning sessions
- Establish naming standards
- Analyze performance

Figure 4-4. IBM MQ administrative tasks

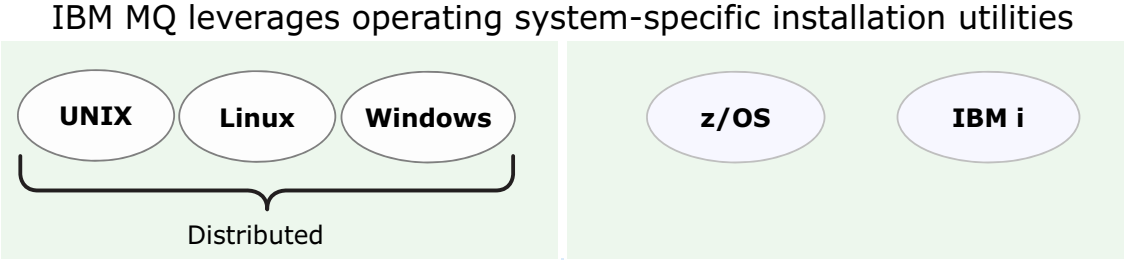
An MQ administrator is responsible for the installation, configuration, and maintenance of MQ and any MQ add-in options such as MQ Managed File Transfer. Configuration tasks include creating queue managers, queues, clusters, and channels. They are also responsible for administering publish/subscribe topologies, implementing security, completing system back ups, monitoring the resources, and identifying problems.

MQ administrators might also spend significant time in meetings, troubleshooting efforts, and helping others.

Ideally the MQ administrator should be involved in infrastructure planning sessions. It is important to keep the MQ administrator aware of any new capacity requirements for new or updated applications.

IBM Training

IBM MQ installation



- Operating system utilities are used to set up file spaces, networking capabilities, and user ID and group administration
- Distributed operating systems leverage installation packages such as `pkgadd`, `smit`, and `rpm`
- MQ control commands are provided to manage multiple version installations

- z/OS uses standard SMP/E facilities to create the MQ libraries
- Comprehensive batch utilities are provided to set up needed z/OS data sets
- IBM i leverages the RSTLICPGM utility for installation

Figure 4-5. IBM MQ installation

The process to install MQ depends on the operating system. MQ uses many of the standard utilities that are available on the operating system to install the product. On distributed operating systems such as UNIX, Linux, and Windows, MQ control commands are used to manage multiple version installations.

MQ uses SMP/E for installation on z/OS. SMP/E validates the service levels and prerequisite and corequisite products, and maintains the SMP/E history records to record the installation. It loads the MQ for z/OS libraries and checks that the loads are successful.

MQ on IBM i uses the RSTLICPGM utility for installation. On MQ for IBM i, you install the MQ server in its primary language, installing samples, and installing extra languages.

IBM Training

IBM MQ control commands

- Entered from the shell or command line
- Require that the user who enters them have proper MQ authorization
- Three categories:
  - Queue manager commands
  - Channel commands
  - Utility commands
- Command syntax, use, and options are documented in the IBM Knowledge Center for MQ
- MQ Explorer has equivalent functions for many of the control commands

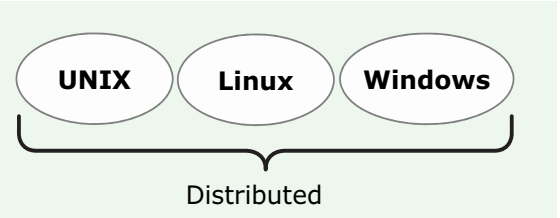


Figure 4-6. IBM MQ control commands

On distributed operating systems such as UNIX, Linux, and Windows, the administration can use MQ control commands to create and manage MQ components.

On Windows, all control commands can be entered from a command line. On UNIX and Linux systems, all MQ control commands can be entered from a shell.

MQ administrators can use all MQ commands, which include the commands to grant MQ authorities for other users. An MQ administrator must be a member of a special group that is called the `mqm` group.

Control commands are categorized as queue manager command, channel commands, and utility commands.

A subset of the control commands can be entered by using the MQ Explorer.

IBM Training

Examples of IBM MQ control commands

Command	Description
`crtmqm`	Create a queue manager
`strmqm`	Start a queue manager
`endmqm`	Stop a queue manager
`dspmq`	Display all queue managers on a server
`dspmqinst`	Display all MQ installations on a server
`dspmqver`	Display the MQ version
`mqrc`	Provide a text interpretation of a numeric error code
`dmpmqcfg`	Back up queue manager object definitions
`runmqsc`	Process MQ script (MQSC) commands

Figure 4-7. Examples of IBM MQ control commands

This figure lists examples of some MQ control commands.

These commands can be entered directly into a command interface or can be packaged in scripts. For example, an administrator can create a script backs up all queue managers on a server.

The `crtmqm`, `strmqm`, and `endmqm` commands are queue manager commands that create, start, and stop a queue manager. It would be expected that only an MQ administrator should handle these tasks, especially in a production environment.

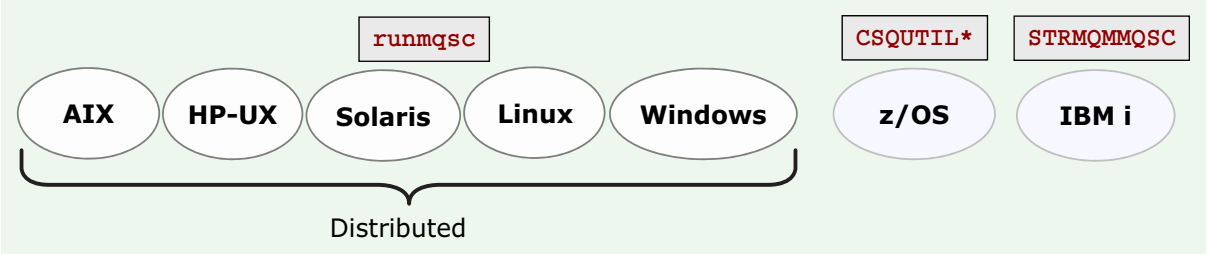
The rest of the commands in the figure are utility commands that are used for maintenance and problem determination.

The last command in the figure, `runmqsc`, is a special command that starts the MQ script (MQSC) command interface on distributed operating systems.

IBM Training

IBM MQ script (MQSC) commands

- Used to administer the queue manager and queue manager objects such as queues, channels, subscriptions
- Enter interactively in the local server, or submit in batch mode by using an operating system utility



- `runmqsc` supports three modes:
  - Direct: Commands are run on a local queue manager
  - Indirect: Commands are run on a remote queue manager
  - Verification: Commands are checked for syntax but not run
- MQ Explorer has equivalent functions for most MQSC commands

Figure 4-8. IBM MQ script (MQSC) commands

The administrator can use MQSC commands to manage queue manager objects, which include the queue manager itself, queues, process definitions, channels, client connection channels, listeners, services, clusters, and authentication information objects.

The `runmqsc` control command starts the MQSC command interface on distributed operating systems in one of the following command modes.

- In **Direct** mode, commands run against the specified local queue manager.
- In **Indirect** mode, commands run against the specified remote queue manager.
- In **Verification** mode, commands are checked for syntax but are not run.

On z/OS, MQSC commands can be entered by using the CSQUTIL batch utility.

Many of the functions of the MQSC commands are also supported in MQ Explorer; however, MQSC commands can be included in scripts to automate repetitive tasks.

IBM Training

Examples of MQSC commands

- Some MQSC commands are specific to an MQ product or operating system
- Command syntax, use, and options are documented in the IBM Knowledge Center for MQ

- Queues

```
DEFINE QLOCAL
ALTER QLOCAL
DELETE QLOCAL
DISPLAY QLOCAL
DISPLAY QSTATUS
CLEAR QLOCAL
```

- Publish/subscribe

```
DEFINE TOPIC
DEFINE SUB
DIS SBSTATUS
DIS PUBSUB
```

- Channels

```
DEFINE CHANNEL
START CHANNEL
STOP CHANNEL
DISPLAY CHANNEL
DISPLAY CHSTATUS
```


Figure 4-9. Examples of MQSC commands

This figure shows some examples of MQSC commands for managing queues, publish/subscribe, and channels:

- The `DEFINE` commands create an object.
- The `ALTER` commands modify an object.
- The `DELETE` commands remove an object.
- The `CLEAR` command removes all messages from a queue.
- The `DISPLAY` commands provide details about an object.

IBM Training

Programmable Command Format (PCF) commands

- Provides a way to automate local and remote administration tasks programmatically
- Same functionality as MQSC commands but not human-readable

Example: Using **Inquire Queue** PCF for remote administration

1. Complete requirements:
   - Developer writes an MQI program and selects the **Inquire Queue** PCF
   - Target queue manager command server component is active
   - Channels exist between the source and target queue manager
2. Program sends an MQPUT with the **Inquire Queue** PCF from the source queue manager to the target queue manager
3. Target queue manager processes the command and replies to the source queue manager, which receives the message with an MQGET

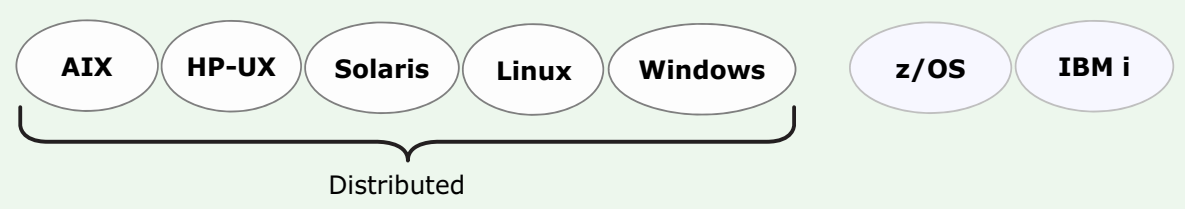


Figure 4-10. Programmable Command Format (PCF) commands

Each queue manager has an administration queue with a standard queue name. Each queue manager also has a command server to service the command messages from the administration queue. Programmable Command Format (PCF) defines command and reply messages that can be sent to any queue manager administration queue in a network.

PCF commands and reply messages are sent and received by using the MQI.

PCF commands provide a way to automate local and remote administration task on IBM i, UNIX, Linux, Windows, and z/OS. For example, an administrator can use the **Inquire Queue** PCF command to get information about a queue.

IBM Training

IBM MQ Administration Interface (MQAI)

- Programming interface to MQ that uses the C language and Visual Basic (on Windows)
- Simpler MQI alternative for sending and receiving PCFs
- Same MQ administration that is done by sending PCF messages to the command server and waiting for a response

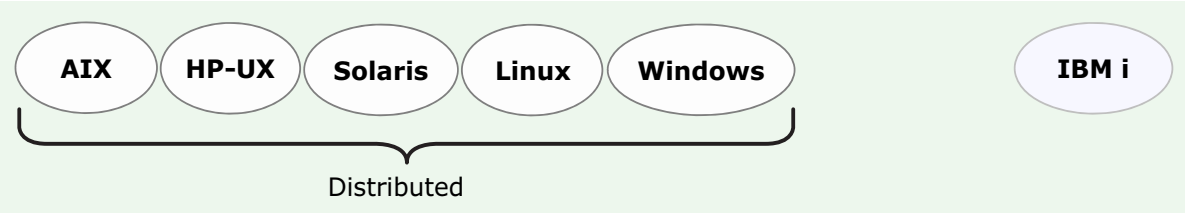


Figure 4-11. IBM MQ Administration Interface (MQAI)

IBM MQ Administration Interface (MQAI) is a programming interface for MQ administration. It uses data bags to handle properties of objects.

The MQAI offers easier manipulation of PCFs than using the MQ MQGET and MQPUT calls.

The MQAI programming interface can use C language and Visual Basic for Windows. It is available on all operating systems except z/OS.

You can use the MQAI to simplify the use of PCF messages, handle error conditions, and exchange data between applications.

IBM Training

IBM MQ Explorer

- Administration tool for local and remote administration
- Provides equivalent functions to many of the MQ command sets
- Available on Windows and Linux x86
- Good option for most administration requirements for developers and business users

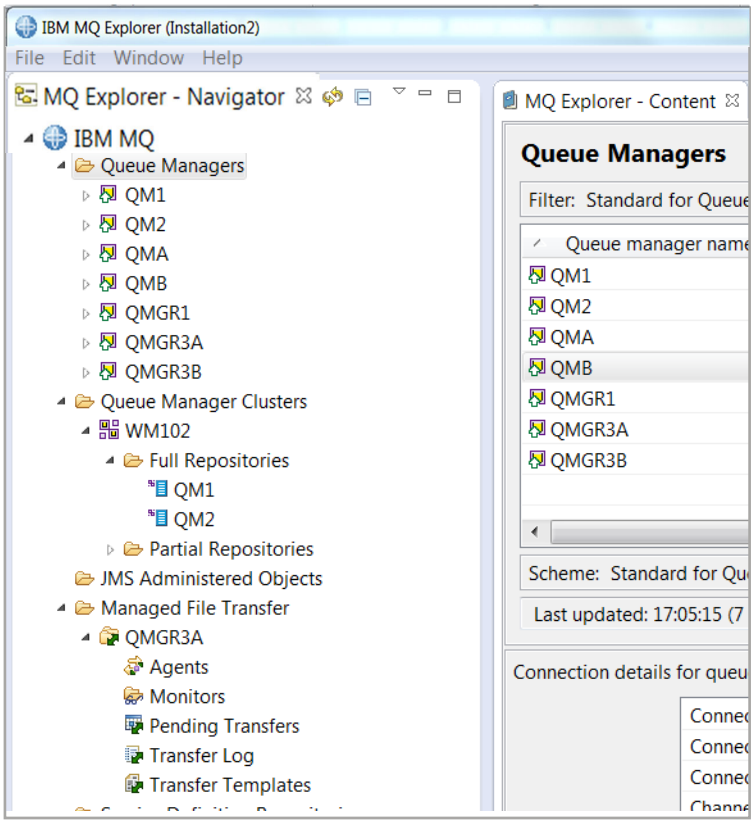


Figure 4-12. IBM MQ Explorer

MQ Explorer is the graphical user interface for administering and monitoring MQ objects, whether they are hosted by a local computer or on a remote system.

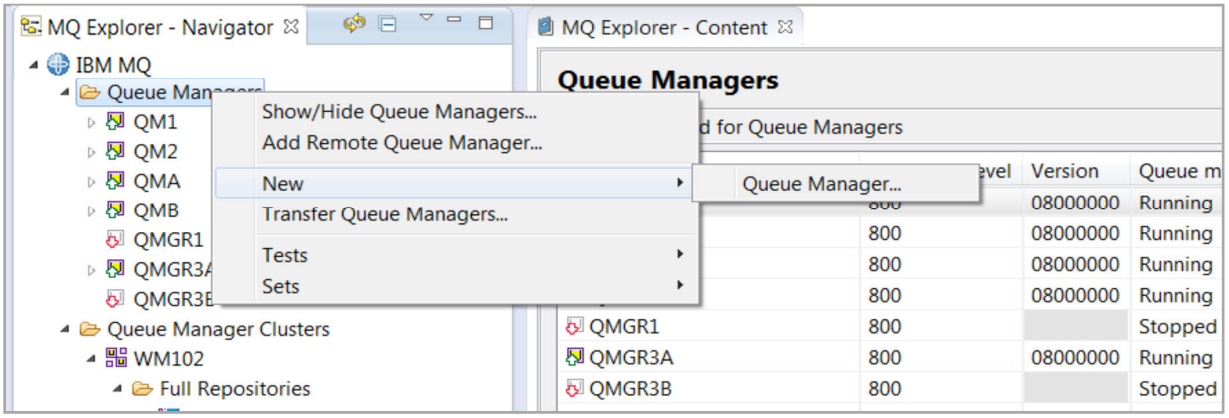
MQ Explorer runs on Windows and Linux x86-64. It can remotely connect to queue managers that are run on any supported operating system, which enables the entire messaging backbone to be viewed, explored, and altered from the console.

MQ Explorer offers an intuitive, comprehensive solution to not only accessing queue managers but administering managed file transfer, telemetry, and JMS. MQ Explorer is a valuable tool for technical and business users.

Using MQ Explorer removes the need to create a server ID for everyone who needs to see whether messages arrived in a queue.

IBM Training

IBM MQ Explorer: Queue manager administration



- **Queue Managers** view shows all local and remote queue managers
- Non-critical queue managers can be removed from view
- Can define new local queue manager (distributed platforms)
- Selecting individual queue managers lets you start and end the queue manager

Figure 4-13. IBM MQ Explorer: Queue manager administration

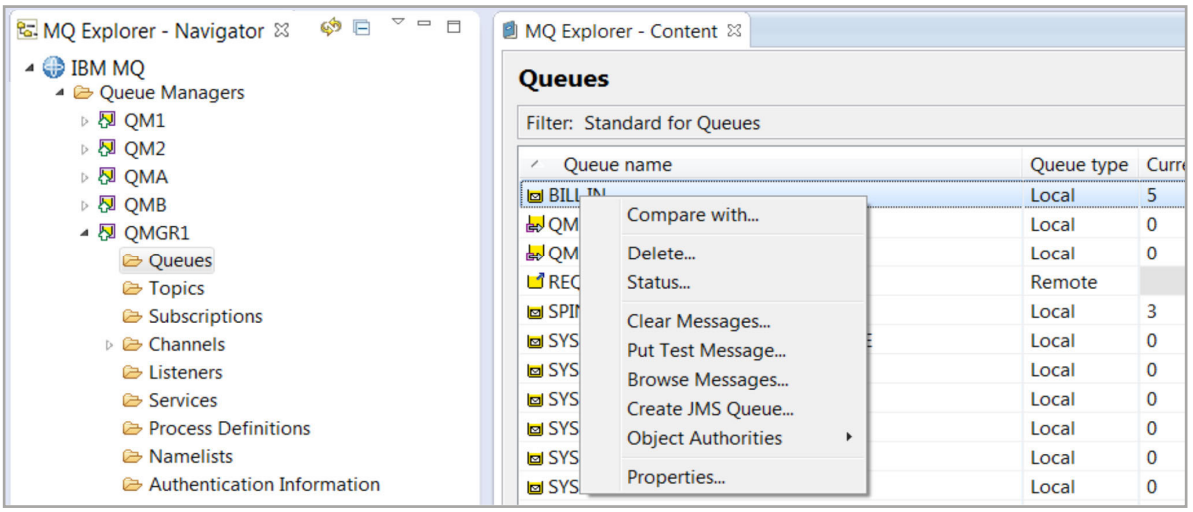
Users with the proper authority can create, delete, and modify queue managers.

A status icon indicates the current state of each queue manager. In the example, queue manager QM1 is running as indicated by the green arrow. Queue manager QMGR1 is stopped, as indicated by the red arrow.

For ease of management, an administrator can group queue managers into logical sets. For example, queue managers can be identified as production or development queue managers.

IBM Training

IBM MQ Explorer: Queue administration



- Check the number of messages in the queue
- Browse messages in the queues
- View message definition attributes
- Define new queues
- Define object authorities and queue attributes

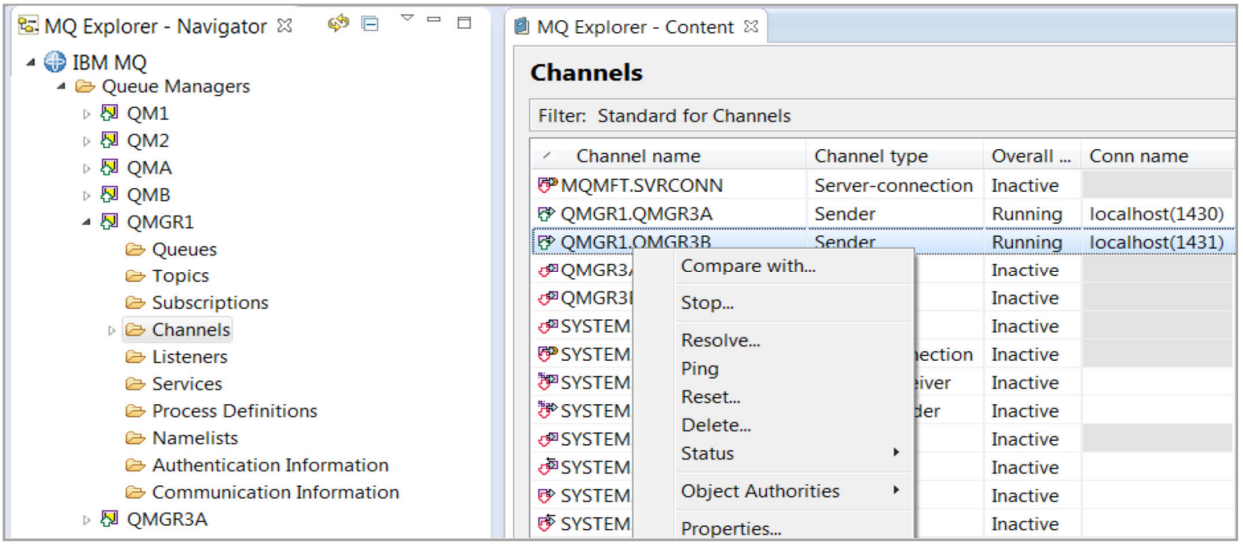
Figure 4-14. IBM MQ Explorer: Queue administration

In MQ Explorer, a user with the proper authority can create, delete, and browse queues and modify queue attributes.

An administrator can also use MQ Explorer to define object authorities for applications and users.

IBM Training

IBM MQ Explorer: Channel administration



- Check the channels status
- Start, stop, and test a channel
- View and modify channel definition properties
- Define a channel

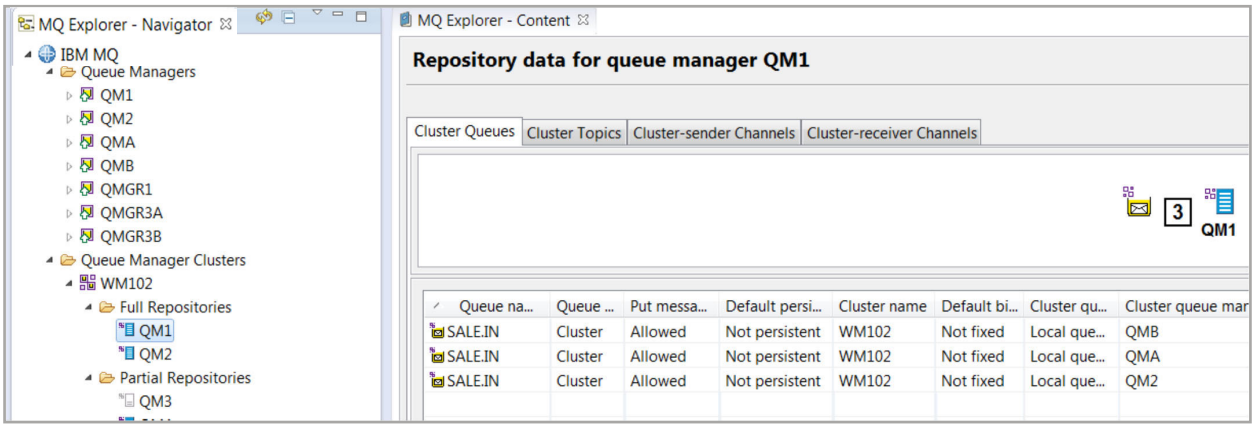
Figure 4-15. IBM MQ Explorer: Channel administration

In MQ Explorer, an authorized user can administer queue manager and client connection channels. An authorized user can define, start, stop, and test a channel.

A status icon in the `Channels` view indicates the status of the channel.

IBM Training

IBM MQ Explorer: Cluster administration



- Define a cluster
- Add queue managers to the cluster
- Identify full repository queue managers
- Determine the number of queue managers that host a specific queue
- Check the status of the cluster channels

System administration overview

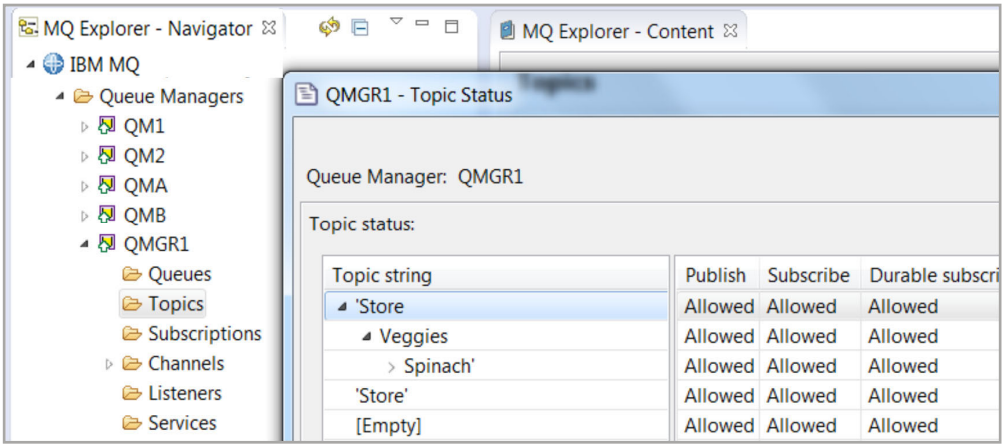
Figure 4-16. IBM MQ Explorer: Cluster administration

MQ Explorer can be used to define and manage queue manager clusters and cluster objects such as cluster queues and channels.

The **Queue Manager Clusters** folder in the **MQ Explorer - Navigator** view lists any known clusters and identifies the full and partial repository queue managers.

IBM Training

IBM MQ Explorer: Publish/subscribe administration



- View the status of the current topic tree
- Define topics and subscriptions
- View the attributes of defined topics or subscribers

Figure 4-17. IBM MQ Explorer: Publish/subscribe administration

MQ Explorer can also be used for publish/subscribe subscriptions and topics.

IBM Training

z/OS ISPF panels

You can control IBM MQ and enter control commands on z/OS by using ISPF panels

Figure 4-18. z/OS ISPF panels

IBM ISPF for z/OS is a multifaceted development tool set for IBM System z that provides host-based software development, including software configuration management.

ISPF for z/OS enables programmers to develop and document batch and interactive programs and administrators and systems programmers to monitor and control program libraries and communications with z/OS. An administrator can use ISPF operations and control panels to manage MQ and MQ objects.

IBM Training

Backup and recovery

Backup

- Allow recovery of queue managers against possible corruption or loss of data that hardware failures cause
- Re-create objects that were deleted or missing
- Simplify administration tasks in MQ including:
  - Migration
  - Cloning systems
  - Automation of application migrations
  - Auditing
- IBM MQ file system backup can be done as part of a full system backup

Recovery

- Two ways of maintaining records of queue manager activities: circular logging and linear logging

Figure 4-19. Backup and recovery

Developing backup and recovery procedures at your site is vital to avoid costly and time-consuming losses of data. MQ provides tools for recovering both queues and messages to their current state after a system failure.

MQ ensures that messages are not lost by maintaining recovery logs of the activities of the queue managers that handle the receipt, transmission, and delivery of messages. It uses these logs for three types of recovery:

- Restart recovery, when you stop MQ in a planned way.
- Failure recovery, when a failure stops MQ.
- Media recovery to restore damaged objects.

IBM Training

Queue manager logs and recovery

- Circular logging
  - 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
- Linear logging
  - Keeps the log data in a continuous sequence of files
  - Can re-create lost or damaged data by replaying log contents (media recovery)
  - Log files are not reused
  - Number of logs can exceed capacity
  - Requires log archival maintenance

Figure 4-20. Queue manager logs and recovery

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

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

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.

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.

Use linear logging if you want both restart recovery and media recovery (re-creating lost or damaged data by replaying the contents of the log).

IBM Training

IBM i journals

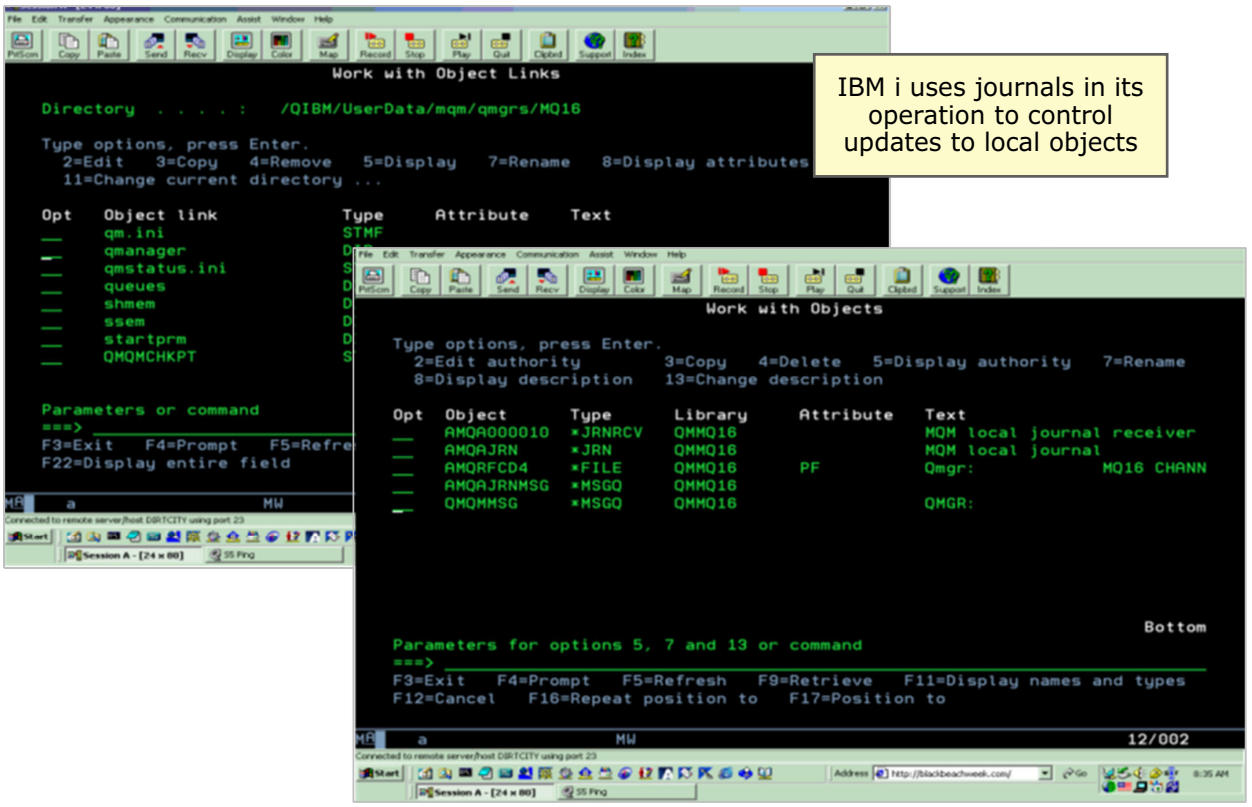


Figure 4-21. IBM i journals

IBM MQ for IBM i holds its data in an individual library for each queue manager instance, and in stream files in the IFS file system.

The queue manager-specific libraries contain journals, journal receivers, and objects that are required to control the work management of the queue manager. The IFS directories and files contain MQ configuration files, the descriptions of IBM MQ objects, and the data they contain.

For MQ on IBM i, two types of MQ backup must be considered:

- Data and journal backup
- Journal backup

As part of your backup strategy, the administrator must take care of the journal receivers. It is useful to remove journal receivers from the MQ libraries for various reasons:

- To release space
- To improve the performance when starting a queue manager
- To improve the performance of re-creating objects

IBM Training

IBM MQ logging considerations

- Logging is critical to the health of a queue manager
- Different roles across an organization influence MQ logging

Infrastructure management and administrators	Infrastructure management architects and developers
• Participate in capacity planning • Monitor logs • If using linear logs, ensure that log archiving procedures are executing correctly • Establish backup and restore procedures • Monitor dead-letter queue	• Obtain projected message size and volume to aid in capacity planning • Establish architectural and development guidelines • Avoid long-running transactions • Use persistence only if needed • Avoid creating multi-message affinities • Do not use MQ as a database

Figure 4-22. IBM MQ logging considerations

The reliability of MQ makes it a trusted component of critical applications. Organizations that establish, communicate, and enforce standards across all stakeholders prevent many unexpected problems.

Architectural standards are key. Architects should understand that they need to obtain projected volume and message sizes, and persistence requirements for the applications they are designing.

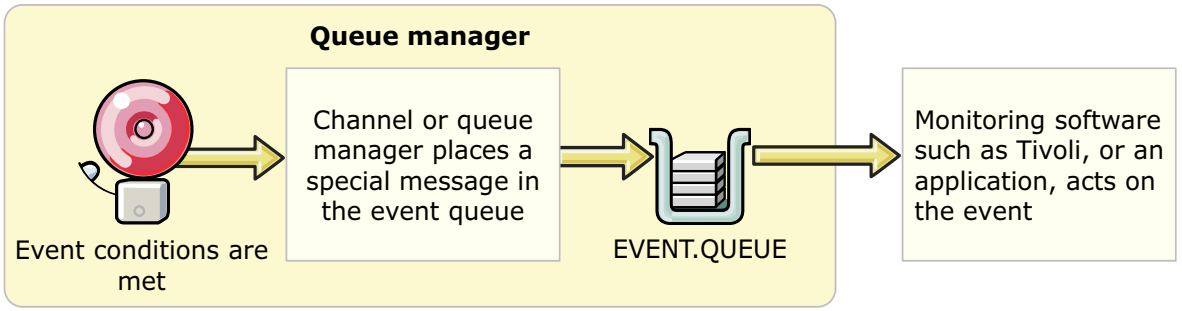
Consider redesigning any process that is expected to run for a long time, such as a complex SAP query, so that MQ does not wait a long time for a response. Long-running processes interfere with the way the log process works and extend the time a log file is marked active rather than being freed for reuse. Long-running processes can be redesigned as callbacks that reinitiate contact with the requester after the response is ready to be returned.

It is important to refrain from using MQ as a repository of data; MQ is not intended to be a database. Messages are kept in queues, and if persistent, they are also kept on the MQ logs until they are processed. Do not design applications to accumulate messages in a queue if they are not going to be removed in a timely manner. Some messages do not warrant being made persistent, such as a query for which the user does not wait, or time-sensitive information.

Regardless of whether your role is that of an MQ administrator or business user, application design without governance might cause outages in the environment that have little to do with MQ administration.

IBM Training

Monitoring and auditing: Instrumentation events



- MQ generates different types of events, such as queue manager, channel, performance, configuration, and logger events
  - Some types of event specific to the operating system
- The administrator enables events
- Events can be directed to a specific queue manager for a single point of control
- Actions taken when an event occurs are handled with packaged monitoring software or a user-written program
- Examples of event conditions are a full queue or an unauthorized MQOPEN

Figure 4-23. Monitoring and auditing: Instrumentation events

MQ events are mechanisms that can be used for auditing and monitoring purposes.

MQ has numerous traceable events. Each event captures different types of details according to the event, and gets written to different event queues also according to type of event. For example, a configuration event is written to the SYSTEM.ADMIN.CONFIG.EVENT queue any time that an MQ object is created, changed, or deleted.

Events are enabled in the queue manager. When the conditions for the event are met, an event message is placed in the corresponding event queue.

After events are generated, it is critical to have a process to get the event messages to prevent queues from filling up with unprocessed events.

IBM Training

Statistics and accounting data

- IBM MQ collects sets of data to be written as messages to predefined queues
  - Data can be post-processed to give information about system activity
  - Data can be used for capacity planning, chargeback, and other information
- Statistical data collection is divided into three categories
  - MQI statistics
  - Queue statistics
  - Channel statistics
- Collection of data for each class can be selected independently
- Queue manager and queue or channel attributes control collection
- Accounting data collects information about MQI applications

Figure 4-24. Statistics and accounting data

Queue managers generate accounting and statistics messages to record information about the MQI operations that MQ applications complete, or to record information about the activities that occur in an MQ system.

- Accounting messages are used to record information about the MQI operations that MQ applications complete.
- Statistics messages are used to record information about the activities that occur in an MQ system.

User applications can retrieve the messages from these system queues and use the recorded information for various purposes:

- Account for application resource use
- Record application activity
- Plan for capacity
- Detect problems in the queue manager network
- Help determine the causes of problems in the queue manager network
- Improve the efficiency of the queue manager network
- Confirm that the queue manager network is running correctly

IBM Training

Performance monitoring

- Display real-time monitoring information for a queue or channel by using IBM MQ Explorer or the appropriate MQSC command
- New in IBM MQ Version 9.0, develop your own program to monitor system resources by using the IBM MQ Performance monitoring API
  - Publishes statistics to a system topic
  - Statistics can be viewed by writing an application that subscribes to the resource monitoring system topic

Figure 4-25. Performance monitoring

A number of monitoring techniques are available in MQ to obtain statistics and other specific information about how your queue manager network is running. Real-time monitoring is a technique that you can use to determine the current state of queues and channels within a queue manager. The information that is returned is accurate at the moment the command was sent.

A number of commands are available to return real-time information about queues and channels. Real-time monitoring can be used in the following tasks:

- Helping system administrators understand the steady state of their MQ system, which helps with problem diagnosis if a problem occurs in the system.
- Determining the condition of a queue manager at any moment, even if no specific event or problem was detected.
- Assisting with determining the cause of a problem in your system.

A new feature in IBM MQ Version 9.0 publishes information messages to a system level topic strings. An authorized user can subscribe to the topics to receive monitoring information for the queue manager and application activity on it. These statistics can be viewed by running the `amqsrua` sample, or by writing an application that subscribes to the resource monitoring system topic.

IBM Training

Unit summary

- Summarize the system administration interfaces for IBM MQ
- Identify basic differences between IBM MQ on distributed operating systems and z/OS
- Contrast administrative interfaces according to role
- Summarize the system administration tasks
- Describe the installation options
- Describe the concepts of logging and recovery
- Summarize the administrative features of IBM MQ Explorer

Figure 4-26. Unit summary

Review questions

1. What are the three categories of IBM MQ control commands?
   A. Channel commands
   B. Utility commands
   C. Trigger commands
   D. Queue manager commands
2. True or False: Scripts with the object definitions can be used as input to the `runmqsc` utility on distributed operating systems.
3. True or False: IBM MQ Explorer provides equivalent functions to most MQSC commands.

Figure 4-27. Review questions

Write your answers here:

1.

2.

3.

IBM Training

Review answers

1. What are the three categories of IBM MQ control commands?
   a. Channel commands
   b. Utility commands
   c. Trigger commands
   d. Queue manager commands
   The answer is A, B, and D.
2. True or False: Scripts with the object definitions can be used as input to the `runmqsc` utility on distributed operating systems.
   The answer is True.
3. True or False: IBM MQ Explorer provides equivalent functions to most MQSC commands.
   The answer is True.

Figure 4-28. Review answers

Unit 5. Security overview

Estimated time

00:30

Overview

This unit summarizes the IBM MQ security requirements and capabilities. It also describes the additional security features that IBM MQ Advanced Message Security provides.

How you will check your progress

- Review questions

References

IBM MQ Library: www.ibm.com/software/integration/wmq/library

IBM Training

Unit objectives

- Summarize the options for secure messaging
- Describe where security is needed across IBM MQ components
- Describe the IBM MQ security mechanisms
- Summarize the features of IBM MQ Advanced Message Security

Figure 5-1. Unit objectives

IBM Training

Terminology checkpoint: Security areas

- **Identification**: Determining the identity of a person or resource that is using a system or resource
- **Authentication**: Proving the persons or resources that are being identified are who they claim to be
- **Access control** or **authorization**: Limiting access to resources to only those people who need it
- **Confidentiality**: Protection of sensitive information
- **Data integrity**: Ability to detect tampering with critical data
- **Auditing**: Ability to determine whether any unauthorized access was done or attempted

**** The terms "access control" and "authorization" are used interchangeably***

Figure 5-2. Terminology checkpoint: Security areas

A baseline of security terms is necessary before proceeding.

Identification refers to determining who implemented a certain action.

Authentication is proving that an entity is who it claims to be, either by providing an ID and password or presenting an SSL certificate. Identity implies accepting that the ID carried is valid by assuming a prior logon with this ID.

Confidentiality deals with keeping the data from access to unauthorized eyes by encrypting this data.

Data integrity and auditing ensures that the data is not tampered with.

IBM Training

Security considerations for IBM MQ

- MQ provides mechanisms to address the different security areas with platform-specific facilities
- MQ resources to be secured fall into three areas:
  - **Administer MQ**: Create queue managers, queues, channels, administer security, work with MQ libraries, and access log files
  - **Use of the MQI**: Connect to a channel, open a queue, and publish or subscribe to a topic
  - **Channel security**: Connect to a queue manager, open a transmit queue or target queues, and administer channel initiators and listeners
- Architectural considerations
  - Application-level security
  - Link-level security

Security overview

Figure 5-3. Security considerations for IBM MQ

Security is an important consideration for both developers of MQ applications, and for system administrators who configure MQ authorities.

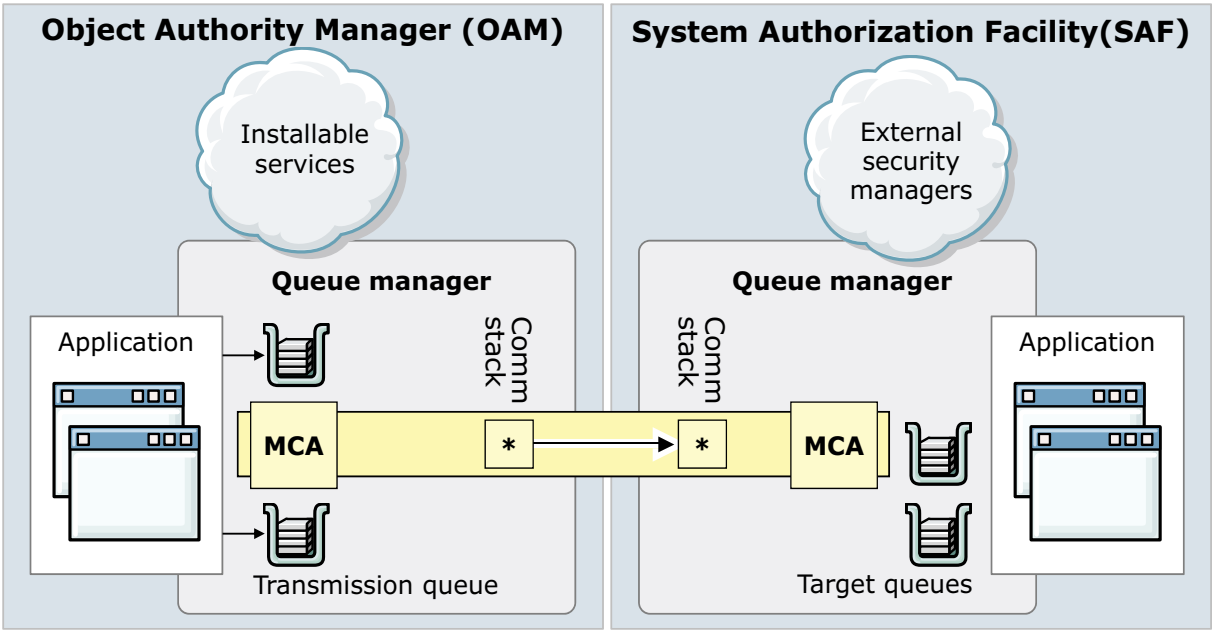
MQ provides mechanisms for securing MQ components and resources, the MQI, and channels. When you are planning an MQ implementation, consider which security mechanisms you require to implement those aspects of security that are important to you.

You must also consider security at the application level and at the link level.

IBM Training

Access control mechanisms

- Two equivalent mechanisms:
  - Secure administration
  - Work with MQ objects



Security overview

Figure 5-4. Access control mechanisms

You can use authorization to limit what particular individuals or applications can do in your MQ environment. Authorizations apply to different MQ objects such as queues, channels, and topics. Authorizations can be grouped under MQI programming or administrative authorizations, or can be specific to provide some users with the ability to browse a queue, for example.

Access control for MQI calls is granted to users or groups with the Object Authority Manager (OAM) for distributed operating systems, or external security manager mechanisms for z/OS.

On z/OS, MQ uses the System Authorization Facility (SAF) to route requests for authority checks to an external security manager such as the z/OS Security Server Resource Access Control Facility (RACF).

Examples of access control:

- When an application sends an MQI “Connect” call, the queue manager checks whether the ID is allowed to connect to the queue manager.
- When an application sends an MQI “Queue Open” call, the queue manager queries the operating for the ID associated with the application. The queue manager then checks whether the ID is allowed to open the queue.
- When an application sends an MQI “Subscription” call, a security check is made to ensure that the application has authorization to access the topic.

Access control to administer IBM MQ

- Operating system access control
  - MQ administrative ID has the access that is required to administer MQ
- Operating system authorizations
  - Control access to MQ files and data sets
  - Use the external Security Manager utilities on z/OS
- Typical MQ administration tasks:
  - Configure queue managers
  - Enter MQ script (MQSC) commands to create objects such as queues, channels, and topics
  - Assign adequate access control to MQ objects by using OAM or SAF mechanisms

Figure 5-5. Access control to administer IBM MQ

MQ administrators need authority to complete various tasks. This authority is obtained in different ways on different operating systems.

For example, on Windows, Linux, and UNIX an MQ administrator must be is a member of the MQ `mqm` group. Members of this group can access all the MQ resources and can enter MQ control commands.

An administrator can also grant specific authorities to users to complete the following tasks:

- Enter commands to administer MQ
- Use the MQ Explorer
- Use the operations and control panels on z/OS
- Use the MQ utility program, CSQUTIL, on z/OS
- Access the queue manager data sets on z/OS

IBM Training

Identification mechanisms: Context

A user logs on to a server to start an application:

- Operating system is interrogated to obtain the ID of the user that is starting the application
  - This user is considered authenticated, as a logon was originally done to the system where the application is running
- **Identity Context** in MQMD provides the ID of the user who started the application
- **Origin Context** in MQMD provides information about the message such as the sending application

```
****Message descriptor****
  StrucId  : 'MD  '  Version : 2
…
** Identity Context
  UserIdentifier : 'nomad789    '
…
ApplIdentityData : '                          '
  ** Origin Context
  PutApplType    : '11'
  PutApplName    : 'ls\c\Samples\Bin\amqsput.exe'
  PutDate  : '20160418'    PutTime  : '20035997'
```

Security overview

Figure 5-6. Identification mechanisms: Context

In MQ, you can implement identification and authentication by using message context information and mutual authentication.

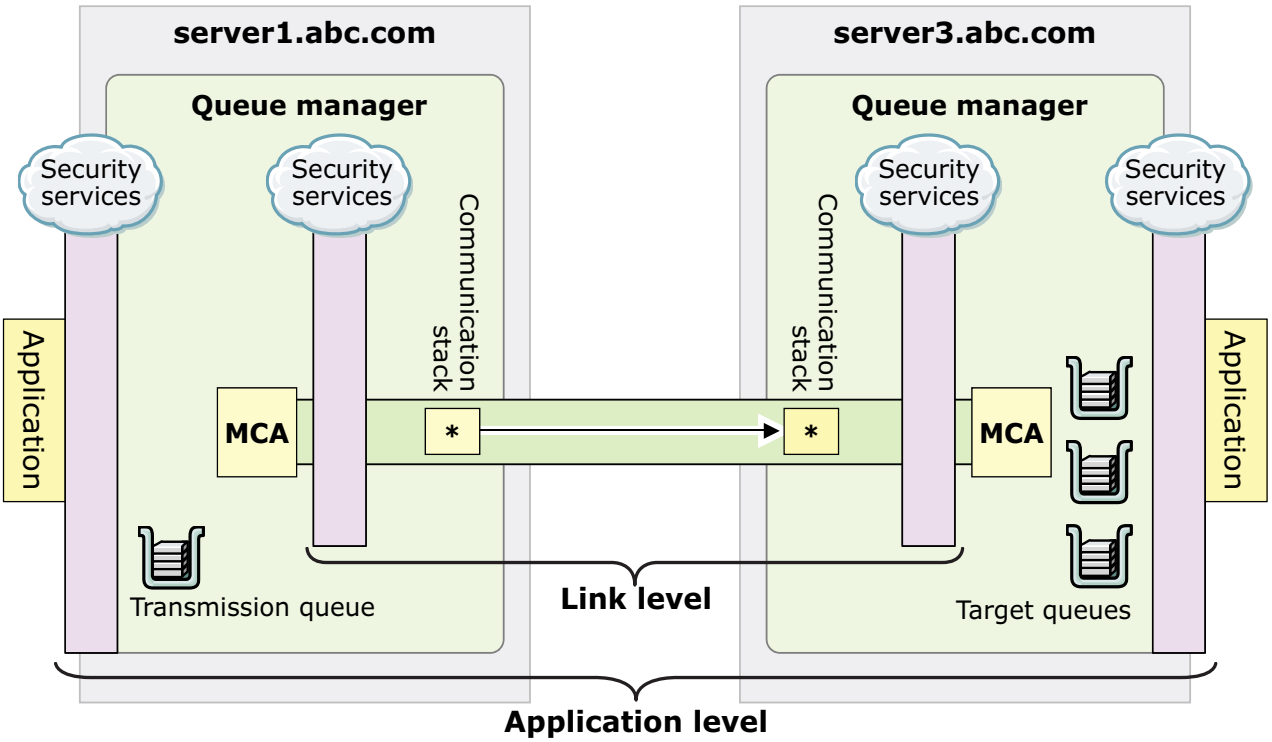
Every message can contain message context information. The context information in a message allows the receiving application to find out about the originator of the message. It contains, for example, the name of the application that put the message and the user ID associated with the application.

The identity and origin context information is held in the MQMD (shown in the figure). It can be generated by the queue manager when a message is put on a queue by an application. Alternatively, the application can supply the information if the user ID associated with the application is authorized to do so.

When a message is first put to MQ, it carries the ID of the person who was logged on to the system. MQ does not authenticate this user; it assumes that the person had to log on to the system, and accepts the ID to identify who put it to the queue. If this message was put on a remote queue manager, the ID would also be that of the person who was logged on and ran the program that placed the message in the queue.

IBM Training

Security at the link and application levels



Security overview

Figure 5-7. Security at the link and application levels

Two terms that are often used when describing MQ security are *link-level security* and *application-level security*.

- Link-level security applies to messages while they are transferred from one queue manager to another. Link-level security involves the security services that are requested by the communications subsystem, the MCA, or a combination. Examples of link-level security are an MCA authenticating its partner, or the encryption or decryption of a message at the sending and receiving end.
- Application-level security, sometimes referred to as *end-to-end security*, applies to protection of messages while they are in queues, also called data at rest. Application-level security refers to data at rest. Examples of application-level security would be encrypting messages that are in a queue, or ensuring the messages that arrived are not tampered with.

IBM Training

Link-level security: Identification and authentication

Channel authentication records

Security mechanism that enables setting rules to determine which connections are allowed and which connections are banned from accessing the queue manager.

- Rules can be set to allow or ban connections, privileged users, or a combination of attributes
- Precedence of rules is important and can be listed
- Rules can be set to use different details of the incoming connection, such as an IP address or a remote queue manager name

Figure 5-8. Link-level security: Identification and authentication

Link-level security refers to those security services that are started, directly or indirectly, by an MCA, the communications subsystem, or a combination of the two working together.

To exercise more precise control over the access that is granted to connecting systems at a channel level, you can use *channel authentication records*.

These rules can specify whether connections are allowed or blocked. If the connection is allowed, the rules can provide a user ID for the channel or use the channel user ID from the client or a user ID that was defined on the channel definition.

Channel authentication records can be created to perform the following functions:

- To block connections from specific IP addresses.
- To block connections from specific user IDs.
- To block connections that claim to be from a certain queue manager unless the connection is from a specific IP address.
- To block connections that present a certain SSL or TLS certificate unless the connection is from a specific IP address.

IBM Training

Link-level security: MCAUSER

- ID used by the MCA
- Which ID is used can vary
- Examples of factors that affect the user ID that is used:
  - When a channel authentication record exists, it takes precedence over the MCAUSER setting
  - If MCAUSER channel definition attribute is not blank, the value in MCAUSER is taken
  - If the channel definition attribute is blank, the channel might use the ID that started the channel
  - Under certain situations, it might pick up the Context ID from the message descriptor

Figure 5-9. Link-level security: MCAUSER

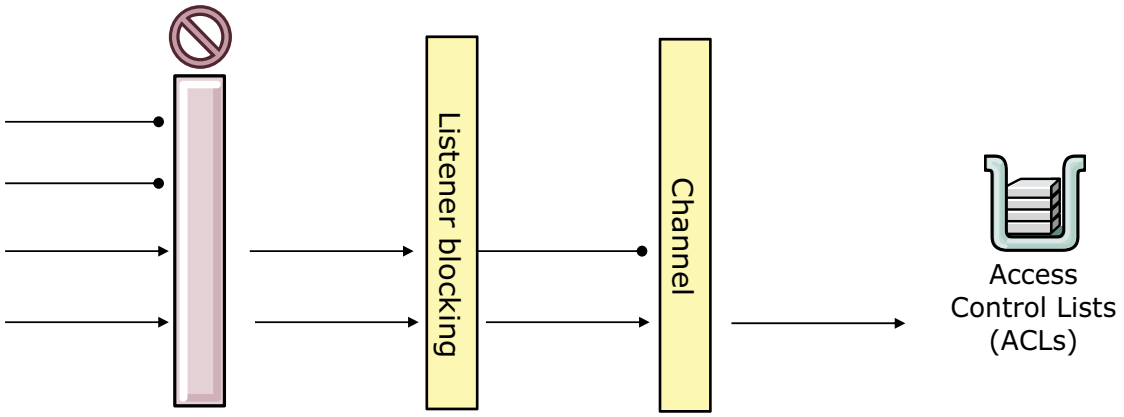
The MCAUSER is a channel attribute that can be used to indicate a specific user whose security profile dictates what authorizations are given to the user of the channel. The ID included in MCAUSER dictates the rights that are granted to users or groups that use this channel.

For example, if a user needs “read only” access to a queue, the administrator can define a user ID and give this user the required MQ authorizations. Then, this user ID is included in the channel definition. If the user tries to remove the messages from the queue, or put messages to the queue, the action fails with a security error.

When the MCAUSER of a channel is left blank, authorization for the channel defaults to the ID of the process that is running the channel, which is a privileged (all access) ID.

IBM Training

Channel blocking points



Listener blocking

- Not a replacement for an IP firewall
- Channel authorization IP blocking rules are checked before any data is read from the socket
- Temporary use only while the firewall is set up

Channel blocking

- Various combinations to block channel access
- Sets MCAUSER as requested in the rule

Figure 5-10. Channel blocking points

MQ can block channels at different points in the connection. This figure illustrates the different blocking points that an inbound connection needs to pass before the connection can get to the queues.

The first blocking point is the IP firewall. MQ security features do not supersede the functions that the IP firewall provides.

The next blocking point is listener blocking. At this point, MQ checks the channel authorization rules.

The final blocking point is channel blocking. At this point, MQ checks the rules and can optionally set the MCAUSER.

IBM Training

Channel authentication rules

- Set rules to control how inbound connections are treated
  - Inbound clients
  - Inbound queue manager to queue manager channels
  - Other rogue connections that are causing FDCs
- Rules can be set to:
  - Allow a connection
  - Allow a connection and assign an MCAUSER
  - Block a connection
  - Ban privileged access
  - Provide multiple positive or negative SSL Peer Name matching
- Rules can use any of the following identifying characteristics of the inbound connection
  - IP address or host name
  - SSL/TLS subject's Distinguished Name
  - Client asserted user ID
  - Remote queue manager name

Security overview

Figure 5-11. Channel authentication rules

Channel authentication rules define authentication handling for inbound connections to the queue manager. Inbound connections include client channels and queue manager to queue manager channels.

These rules can be used to do the following actions:

- Set up appropriate identities for channels to use when they run against the queue manager
- Block unwanted connections
- Ban privileged users

IBM Training

Authentication mechanisms: Security exits

- MQ provides exit points to extend the functions of MQ for messages and channels

- MQ security exit details:
  - Main objective is to allow MCA to authenticate its partner
  - Is a user written program
  - Exit name is included as an attribute of the channel definition
  - Initiated by the channel
  - Is called after the initial data negotiation but before messages flow
  - Work in pairs at the sending and receiving of queue managers

Consider use of the channel authorizations before attempting to implement a security exit.

Figure 5-12. Authentication mechanisms: Security exits

If you require security beyond the functions that MQ provides, you can use MQ security exits to access custom code.

As MQ evolved and improved, the need for security exits lessened. A summary of security exits is provided in this course for completeness. Also, if you are responsible for supporting older versions of MQ, you might find that the implementation includes security exits to provide functions that are now provided by MQ.

IBM Training

Link-level security: Connection authentication

- Queue manager attribute
- Two types of authentication objects
  - OS repository
  - LDAP
- Initial settings:
  - New queue managers default to using OS system authentication
  - Connection authentication disabled on migrated queue managers

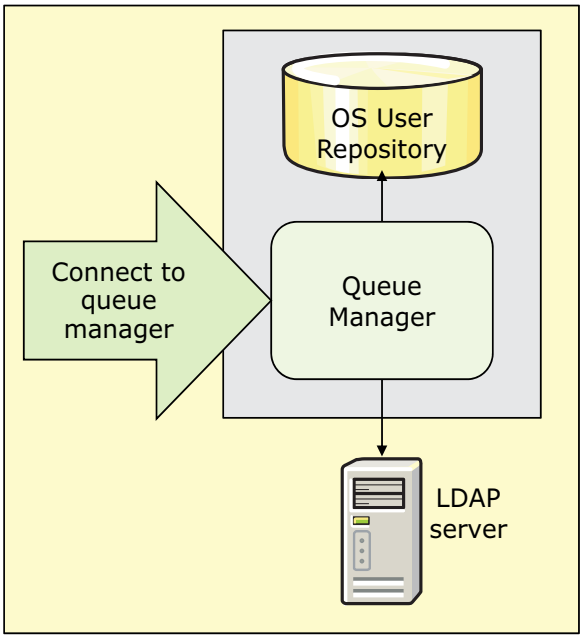


Figure 5-13. Link-level security: Connection authentication

A queue manager can be configured to use a supplied user ID and password to check whether a user has authority to access resources.

MQ can authenticate a connection in various ways:

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

In the middle of the network is the queue manager. The queue manager is responsible for managing resources and the checking of authority to those resources. MQ has many resources that an application might require authority to, but this example uses a queue that must be opened for output.

The user IDs and passwords can be stored and referenced on an LDAP server or an operating system repository.

IBM Training

Link-level security: Confidentiality, data integrity, and authentication

Secure Sockets Layer (SSL)

Using SSL with MQ:

1. Conduct plans to determine SSL details
2. Obtain certificates
3. Create a keystore for each queue manager or client
4. Tell the queue manager or client about the keystore
5. Test channels to be protected *without* SSL before adding SSL attributes to the channel definition
6. Enable SSL on both sides of the connection

Figure 5-14. Link-level security: Confidentiality, data integrity, and authentication

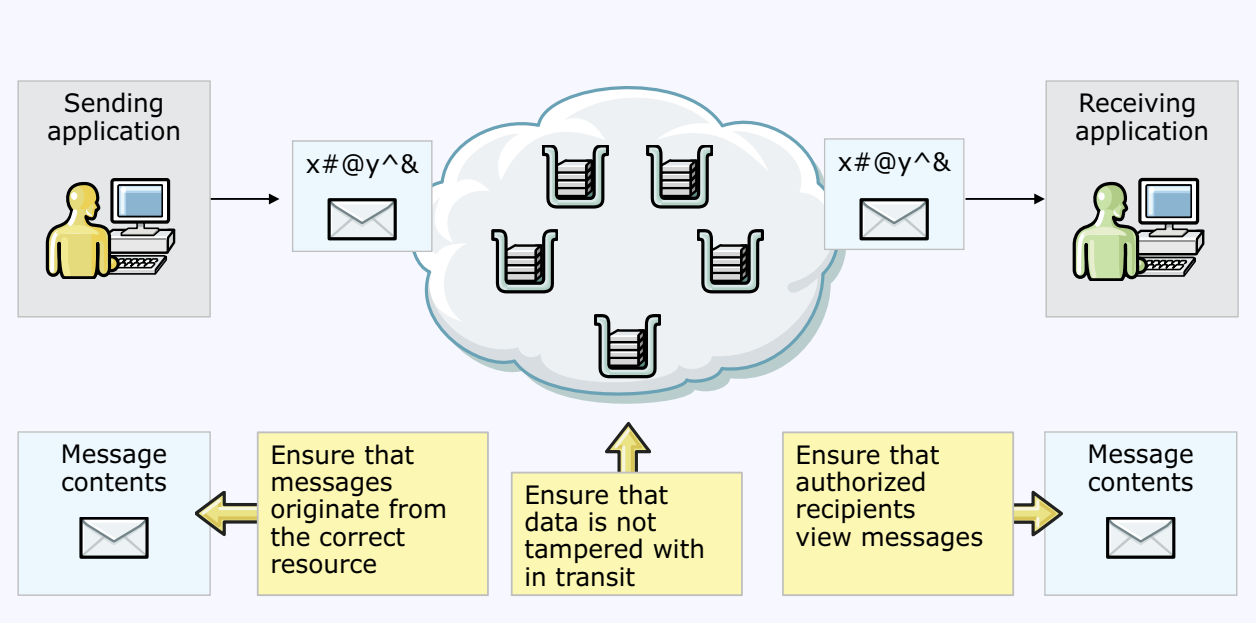
MQ channels can be configured to use SSL to encrypt and authenticate. This figure summarizes the steps for using SSL with MQ.

1. When an organization decides to use SSL, it is important to identify a team and process to manage the certificates, as expired certificates might cause connections to fail.
2. To implement SSL, it is key to resolve any logistics before proceeding. Determine what certificate authority is used, and who manages the certificates. If an organization decides to use SSL in production, it should also implement SSL in its test environment.
3. To implement SSL, an initial configuration is completed in the queue manager to create a keystore and obtain a certificate. Each queue manager must have its own certificate.
4. Assuming that the partner channel is also set up, when a queue manager is configured to use SSL, adding it to a channel is just a matter of updating some of the channel attributes.
5. It is best to test the channel connectivity before SSL is added to the channel. Test the channel first. If you identify SSL problems, it is best to establish first that the channel is not the problem.
6. After you ensure that the channel configuration is correct without SSL, enable SSL on server and client.

IBM Training

Application-level security: Confidentiality and data integrity

- Application-level security pertains to the safety of messages while not in transit, also referred to as “data at rest”



Security overview

Figure 5-15. Application-level security: Confidentiality and data integrity

IBM MQ Advanced Message Security provides the functions necessary to address application-level security concerns.

MQ Advanced Message Security ensures that messages originate from the correct resource, that data is not tampered with in transit, and the authorized recipients can view messages.

IBM Training

IBM MQ Advanced Message Security

- End-to-end, message-level security that offers data protection for your point-to-point messaging infrastructure
- Data encryption and authentication that provides security-rich data transport throughout the messaging cycle
- Centralized policy capabilities that help enforce message security standards and aid compliance
- Integration with IBM MQ and other products for easy use with your existing messaging backbone

Figure 5-16. IBM MQ Advanced Message Security

MQ Advanced Message Security is a separately licensed component of MQ that provides a high level of protection for sensitive data that flows through the MQ network, while not impacting the end applications.

Advanced message security provides the following capabilities:

- End-to-end data protection that uses either encryption or digitally signing messages
- Supports MQ servers or clients
- Uses public key infrastructure to provide authentication, authorization, confidentiality, and data integrity function for messages
- Security policy administration point

MQ applications can use MQ Advanced Message Security to send sensitive data, such as high-value financial transactions and personal information, with different levels of protection by using a public key cryptography model. MQ Advanced Message Security associates users and applications with X.509 standard digital certificates.

In MQ V9.0, MQ Advanced Message Security was redesigned to use an alternative crypto-library, which is built into the IBM MQ classes for Java and IBM MQ classes for JMS. This redesign makes it easier to integrate with your existing messaging backbone.

IBM Training

IBM MQ Advanced Message Security qualities of protection

- Integrity protection
  - Provided by digital signing, which provides assurance on who created the message, and that the message was not altered or tampered with
- Privacy protection
  - Provided by a combination of digital signing and encryption.
  - Encryption ensures that message data is only viewable to the intended recipient, or recipients
  - Even if unauthorized recipients obtain a copy of the encrypted message data, they are unable to view the actual message data itself.
- Confidentiality protection
  - Provided by encryption
  - Allows for symmetric key reuse over a sequence of messages, which can significantly reduce the costs that are involved in encrypting a number of messages that are intended for the same recipient or recipients

Figure 5-17. IBM MQ Advanced Message Security qualities of protection

MQ Advanced Message Security provides three levels of protection

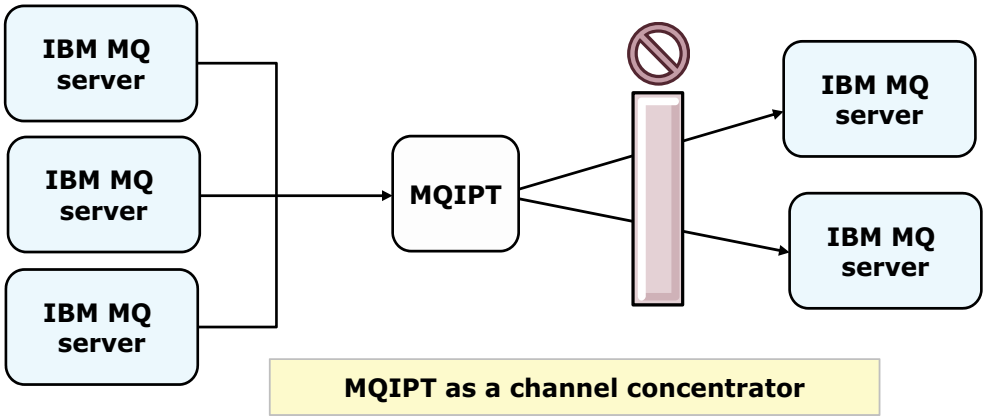
- *Integrity protection* is provided by digital signing, which provides assurance on who created the message, and that the message was not altered or tampered with.
- *Privacy protection* is provided by a combination of digital signing and encryption. Encryption ensures that message data is only viewable to the intended recipient, or recipients. Even if unauthorized recipients obtain a copy of the encrypted message data, they are unable to view the actual message data itself.
- *Confidentiality protection*, new in IBM MQ V9.0, is provided by encryption only. Confidentiality policies do allow for symmetric key reuse over a sequence of messages. You can use this approach to significantly reduce the costs that are involved in encrypting a number of messages that are intended for the same recipient or recipients.

  For example, when you put 10 encrypted messages to the same set of recipients, a symmetric key is generated, and then encrypted for the first message, by using an asymmetric key operation for each intended recipient of the message. Based on policy controlled limits, the encrypted symmetric key can then be reused by subsequent messages that are intended for the same recipients. In this example, 90% of the asymmetric key operations can be avoided by both the putting and getting applications by reusing the same key.

IBM Training

IBM MQ Internet Pass-Thru (MQIPT)

- MQ SupportPack MS81 product extension to MQ
- Can be used in several modes
  - Acts as proxy for MQ protocol
  - Supports HTTP, HTTPS, and SOCKS
  - Allows use of HTTP wrappers
  - SSL proxy mode
  - Works with publish/subscribe
  - Concentrator



Security overview

Figure 5-18. IBM MQ Internet Pass-Thru (MQIPT)

IBM MQ Internet Pass-Thru (MQIPT) is an extension to the base MQ product as a category 3 SupportPac (MS81). It can be downloaded from the IBM MQ SupportPac website (www.ibm.com/software/integration/wmq/supportpacs).

MQIPT runs as a stand-alone service. It can receive and forward MQ message flows, between two MQ queue managers or between an MQ client and an MQ queue manager. The instances of MQIPT allow the two MQ systems to exchange messages without needing a direct TCP/IP connection. This capability might be necessary if a firewall does not allow direct TCP/IP connectivity between the two systems.

The figure shows MQIPT acting as a concentrator. By using MQIPT in this way, channels to or from multiple separate hosts can appear to a firewall as if they are all to or from the MQIPT host. A channel concentrator makes it easier to define and manage firewall rules for filtering.

MQIPT can also use SSL to encrypt and decrypt messages that are sent over the Internet.

Auditing

Distributed operating systems

- Queue manager events
- Security failures such as
  - Connection not authorized
  - Open not authorized
  - Subscription not authorized
- Captured as messages that are sent SYSTEM queue
- Events must be processed with an application or a monitoring system
- Other events can also be checked, such as using a configuration event to capture “someone defined a channel in production”

z/OS

- Use external security manager facilities to capture
  - Failed access to resources
  - Successful access to resources
  - Any access or changes to security profiles
- Must be enabled

Figure 5-19. Auditing

Auditing is an important part of security administration.

For example, you can use audit events to capture historical data on a production queue manager. A configuration event can be set to capture any attempts to create, alter, or delete a channel. If configured correctly, this information can be used to track who did the changes.

This figure summarizes the events that need to be enabled and collected for auditing purposes.

Auditing was introduced in Unit 4.

IBM Training

IBM MQ security summary

- Identification
  - Operating system user IDs
  - Context
  - Channel authentication records
  - Channel security exits
- Authentication
  - Original operating system logon
  - MQI connection
  - Channel Security Exits
  - Channel authentication
  - Connection authentication
  - SSL/TLS
- Access control
  - Command security
  - API security
  - “Put” authority on channels
  - Firewalls and MQIPT SupportPac
- Confidentiality
  - IBM MQ Advanced Message Security
  - SSL/TLS
  - Channel exits
  - API exits including clients
- Data integrity
  - IBM MQ Advanced Message Security
  - SSL
- Auditing
  - Events

Figure 5-20. IBM MQ security summary

While MQ provides numerous mechanisms to implement security at many levels, it is critical to reach a good point where security is adequate but not so restrictive that it causes problems with applications and MQ.

Review security regularly to ensure that all needs are met, and future needs are anticipated to allow adequate planning and implementation.

Security can be expensive to implement and maintain. While the need for security should never be taken lightly, be careful not to over-architect complex processes if these processes are not warranted. For example, if messages are picked up immediately, it is not warranted to encrypt data at rest.

Security does involve a small performance footprint. If security is needed, then configure it. Implement it in the test environment first to prevent any unpleasant surprises in production.

IBM Training

Unit summary

- Summarize the options for secure messaging
- Describe where security is needed across IBM MQ components
- Describe the IBM MQ security mechanisms
- Summarize the features of IBM MQ Advanced Message Security

Figure 5-21. Unit summary

Review questions

1. How can an application determine where a message originated?
   A. Obtain the information from the corresponding receiver channel
   B. Install a third-party tracking solution
   C. Check the identity and origin context fields of the MQ message descriptor
   D. Run a TCP/IP trace

2. True or False: Link level security deals with protecting messages while being transferred between queue managers.

3. What MQ security mechanism gives the ability to authenticate with LDAP or an operating system repository?
   A. MCAUSER
   B. Connection authentication
   C. Advanced message security
   D. OAM

Figure 5-22. Review questions

Write your answers here:

1.

2.

3.

IBM Training

Review answers

1. How can an application determine where a message originated?
   A. Obtain the information from the corresponding receiver channel
   B. Install a third-party tracking solution
   C. Check the identity and origin context fields of the MQ message descriptor.
   D. Run a TCP/IP trace

   The answer is C.

2. True or False: Link level security deals with protecting messages while being transferred between queue managers.

   The answer is True.

3. What MQ security mechanism gives the ability to authenticate with LDAP or an OS repository?
   A. MCAUSER
   B. Connection authentication
   C. Advanced message security
   D. OAM

   The answer is B.

Figure 5-23. Review answers

Unit 6. Introduction to IBM MQ Managed File Transfer

Estimated time

00:30

Overview

This unit introduces IBM MQ Managed File Transfer, identifies file transfer scenarios, and summarizes basic configuration.

How you will check your progress

- Review questions

References

IBM MQ Library: www.ibm.com/software/integration/wmq/library

IBM Training

Unit objectives

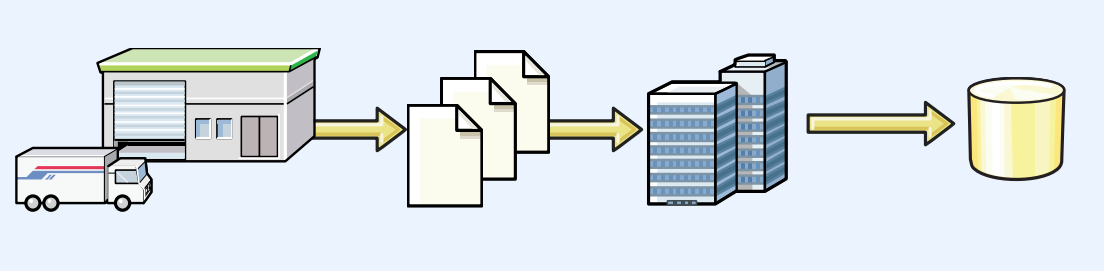
- Identify IBM MQ Managed File Transfer scenarios
- Describe the main components of IBM MQ Managed File Transfer
- Describe IBM MQ Managed File Transfer basic configuration
- Identify various ways of initiating a transfer
- Summarize auditing capabilities for IBM MQ Managed File Transfer

Figure 6-1. Unit objectives

IBM Training

Introducing IBM MQ Managed File Transfer

- Reliable, auditable text, and binary file transfer capabilities
- Transfer files over MQ network or FTP, FTPS, or SFTP
- Transfer files to and from Connect:Direct nodes
- Different choices to transfer files such as:
  - Setting up a transfer schedule with MQ Managed File Transfer mechanisms
  - Manually on demand by using command line or MQ Explorer
  - Placing specially formatted message in agent command queue
  - Using scripts



Introduction to IBM MQ Managed File Transfer

Figure 6-2. Introducing IBM MQ Managed File Transfer

MQ Managed File Transfer is a viable solution for exchanging data for those applications whose payload grows too large to be sent over MQ. It is also a good option for applications that do not have an MQ infrastructure, traditionally use FTP, and need to integrate management and traceability.

IBM Training

IBM MQ Managed File Transfer features

- Complete, flexible file transfer gateway that integrates with your IBM MQ environment
- Reliable and flexible technology that supports fast, cost-effective file transfers
- Complete visibility and auditability of files with central visibility and tracking of all file data movement
- Performance and security-rich file handling capabilities for end-to-end security and encryption of file data
- Integrated administration tools that help to manage and customize your setup

Figure 6-3. IBM MQ Managed File Transfer features

MQ Managed File Transfer is a complete, flexible file transfer gateway. You can use it to exchange files rapidly, securely, and cost-effectively across all major operating systems. You can also use it for most messaging needs, which includes message-to-file and file-to-message, file to file over MQ and peer-to-peer file transfer.

MQ Managed File Transfer provides reliable and efficient transfer of files, regardless of file size or network failure. It protects your sensitive data with secure protocols and encryption and extends MQ to offer more flexible integration options and enrichment with high performance and scalability.

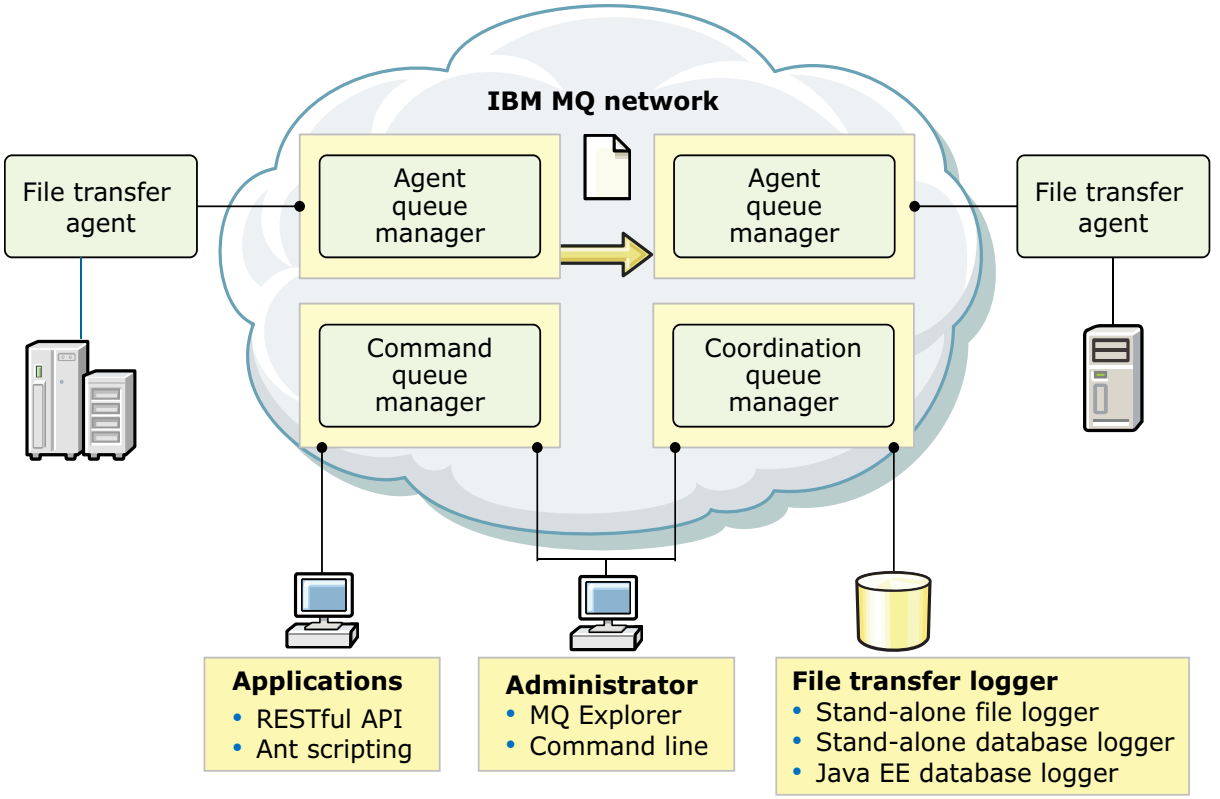
With MQ Managed File Transfer, you can get real-time visibility into file-driven information exchanges within and between businesses. You can get timely and accurate audit trails and reports for file transfer activity.

MQ Managed File Transfer provides high-performance file transfer, capable of managing files of all sizes and volumes. It offers real-time file transfer with fewer reliability issues when compared with FTP transfers.

MQ Managed File Transfer integrates file transfer capabilities directly within the MQ environment without installing a separate product. You can use MQ administration tools such as MQ Explorer to manage file transfers and MQ Managed File Transfer components.

IBM Training

Sample IBM MQ Managed File Transfer architecture



Introduction to IBM MQ Managed File Transfer

Figure 6-4. Sample IBM MQ Managed File Transfer architecture

The figure shows a simple MQ Managed File Transfer topology.

The topology includes two *file transfer agents*, which connect to a dedicated *agent queue manager* in the MQ network. A file is transferred between the file transfer agents through the MQ network.

The MQ network also includes a *coordination queue manager* and a *command queue manager*. Applications and tools connect to these queue managers to configure, administer, operate, and log MQ Managed File Transfer activity in the MQ network.

File transfers can be initiated by application or by an administrator.

When MQ Managed File Transfer transfers files, it publishes information about its actions to a topic on the coordination queue manager. This information can be stored on a file system or in a database. The database logger is an optional component of MQ Managed File Transfer that you can use to copy file transfer information into a database for analysis and auditing purposes.

IBM Training

Product components

- **Coordination queue manager** broadcasts audit and file transfer information
- **Command queue manager** connects to the MQ network and receives MQ Managed File Transfer commands
- **Agent queue manager** works with the agent process to complete the transfer
  - Registers with the coordination queue manager
  - Publishes transfer details to the coordination queue manager
- **File transfer agent** connects to queue manager and transfers file data, as messages, to other file transfer agents
  - Has its own set of queues on its associated queue manager
  - Is attached to its queue manager in either bindings or client mode
  - Can use the coordination queue manager as its queue manager
- **File transfer logger** connects to an MQ queue manager and logs file transfer audit-related data to either a database or a file

Figure 6-5. Product components

MQ Managed File Transfer uses three queue manager roles to control its processes. Each role can be handled with the same or separate queue managers.

- The command queue manager is used to send MQ managed file transfer commands.
- The agent queue manager works with the agent process to complete the transfer, registers with the coordination queue manager, and publishes transfer details to the coordination queue manager.
- The coordination queue manager is the single collection point for agents status, transfer status, and audit information. The coordination queue manager does not need to be active for transfers.

A file transfer agent connects to an MQ queue manager and transfers file data, as messages, to other file transfer agents.

The file transfer logger connects to an MQ queue manager and logs file transfer audit-related data to either a database or a file. The queue manager is often the coordination queue manager.

IBM Training

Product options

- IBM MQ Managed File Transfer Agent
  - Installs file transfer agent
  - Make client or bindings mode connections to queue managers
  - Transfer files to and from other file transfer agents and Connect:Direct nodes
  - Does not require local MQ server
- IBM MQ Managed File Transfer Logger
  - Installs a file transfer logger
  - Requires local MQ server
- IBM MQ Managed File Transfer Service
  - Includes all capabilities of the IBM MQ File Transfer Agent option
  - Transfer files to and from other file transfer agents, SFTP, FTP, or FTPS protocol servers, and Connect:Direct nodes
  - Requires local MQ server
- IBM MQ Managed File Transfer Tools
  - Installs command line tools for starting file transfers, scheduling file transfers, and creating resource monitors from the command line

Figure 6-6. Product options

IBM MQ Managed File Transfer can be installed as four different options, depending on your operating system and overall setup.

- **IBM MQ Managed File Transfer Agent**
  This product option installs a file transfer agent. This product option can be installed on systems that do not contain an MQ server; however, some capabilities of the file transfer agent are only available when an MQ server is installed on the same system. For example, the ability to create and use a protocol bridge requires an MQ server on the same system as the file transfer agent.
- **IBM MQ Managed File Transfer Logger**
  This product option installs a file transfer logger. This option must be installed on a system that contains an MQ server.
- **IBM MQ Managed File Transfer Service**
  This product option installs a file transfer agent that can create a create protocol bridge agents. Protocol bridge agents are used to send and receive files with FTP, FTPS, or SFTP servers. This option must be installed on a system that contains an MQ server.
- **IBM MQ Managed File Transfer Tools**
  This product option installs command tools that are used to interact with file transfer agents. You can use the tools to start file transfers, schedule file transfers, and create resource monitors from a command.

How does it work?

- Transfers files between agent processes by dividing each file into one or more messages and transmitting the messages through the MQ network
- Agent processes move file data by using nonpersistent messages to minimize the impact on MQ logs
- Agent processes regulate the flow of messages containing file data
  - Prevents messages from building up on transmission queues
  - Ensures that if any of the nonpersistent messages are not delivered, the file data is sent again
- Agent can accept specially formatted command messages to instruct the agent
- Agents send information about their state and the progress and outcome of transfers to the coordination queue manager

Figure 6-7. How does it work?

MQ Managed File Transfer transfers files between agent processes by dividing each file into one or more messages and transmitting the messages through the MQ network.

The agent processes move file data by using nonpersistent messages to minimize the impact on the MQ logs. By communicating with one another, the agent processes regulate the flow of messages that contain file data. The agent processes prevent the messages that contain file data from building up on the MQ transmission queues and ensures that if any of the nonpersistent messages are not delivered, the file data is sent again.

MQ Managed File Transfer agents use a number of MQ queues. Although some of these queues are strictly for internal use, an agent can accept requests in the form of specially formatted command messages sent to a specific queue that the agent reads from. Both commands and the MQ Managed File Transfer in MQ Explorer can send MQ messages to the agent.

MQ Managed File Transfer agents send information about their state and the progress and outcome of transfers to the coordination queue manager. This information is published by the coordination queue manager and can be subscribed to by applications that want to monitor transfer progress or keep records of the transfers that occurred. Both the commands and the MQ Explorer can use the information that is published.

IBM Training

Required IBM MQ connectivity paths

- MQ connectivity is required
  - Between agent queue managers for bidirectional communication between the queue managers of agents that exchange files
  - From agent to coordination queue manager
  - From command queue manager to agent queue manager
- MQI client channels are used to connect file transfer agents to the agent, coordination, and command queue managers
- Server-connection channels are required on the queue managers
  - Use default server-connection channel or create a custom channel

Figure 6-8. Required IBM MQ connectivity paths

Define the required MQ infrastructure and the connectivity before configuring MQ Managed File Transfer.

You must define MQI client channels between the file transfer agent and the queue managers that are designated as the agent, coordination, and command queue managers. On the queue manager side of the connection, you can either use the default server-connection channel, or create a custom server-connection channel for the file transfer agent.

MQ Managed File Transfer is simple to implement, but like many products, can be daunting at first. Start with one queue manager for all roles, and defer applying MQ object security until a sample transfer is successfully tested.

IBM Training

Transfer request options: Resource monitors

- File transfer request can be created by configuring a resource monitor to check a directory for the arrival of a file at set poll times

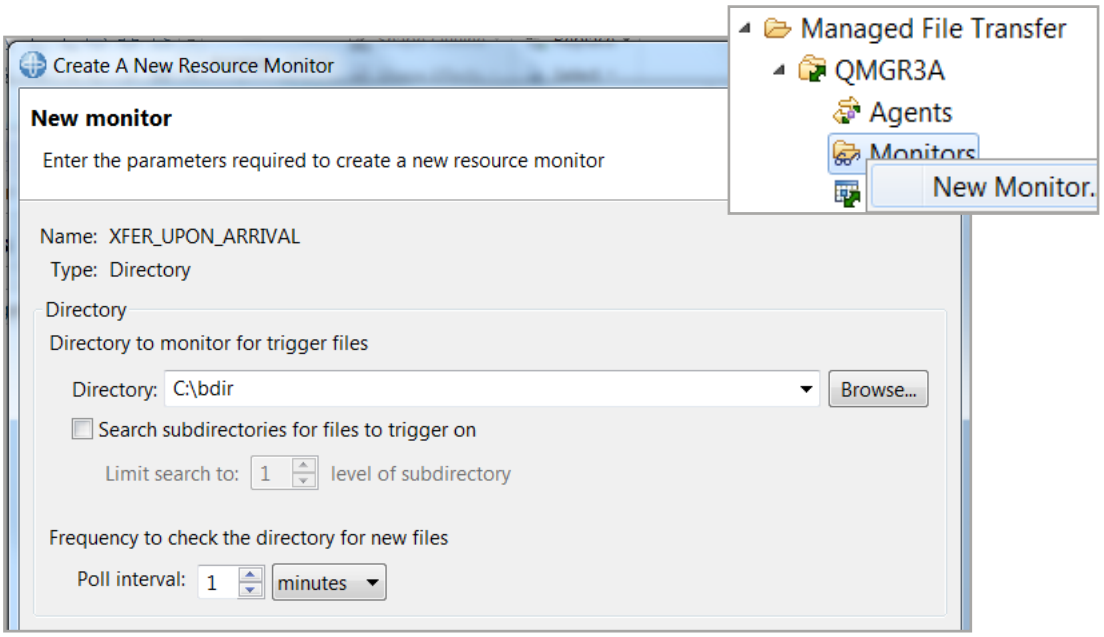


Figure 6-9. Transfer request options: Resource monitors

One option for requesting a file transfer is to set up a monitor to listen for the arrival of a file in a directory. You can set up a monitor in MQ Explorer under the **Managed File Transfer** folder. The **New Monitor** window, which is shown in the figure, also contains an option for scheduling a transfer at certain intervals.

IBM Training

Transfer request options: Command

- File transfer request can be entered in a command line
- Can check results with MQ Explorer **Managed File Transfer > Transfer Log**

```
C:\IBM\MQ\mqft\config\QMGR3A\agents\AGENT3A>fteCreateTransfer -sa AGENT3A
-da AGENT3B -dd C:\adir C:\bdir\toafile.txt
BFGCL0035I: Transfer request issued.  The request ID is:
414d5120514d47523341202
020202020b26b5e5320069503
BFGCL0182I: The request is now waiting to be processed by the agent.
```

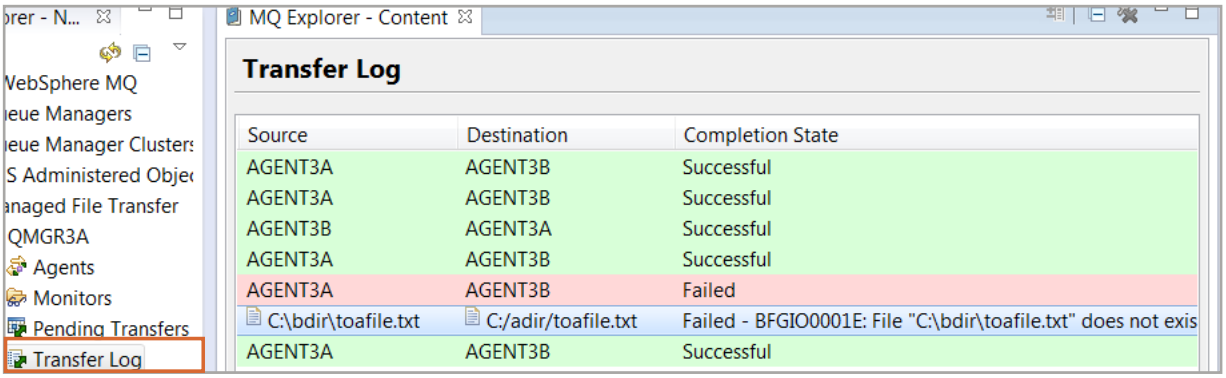


Figure 6-10. Transfer request options: Command

Another option for requesting a file transfer is to use the `fteCreateTranfer` command. This command can also be included in a script to create a scripted transfer.

As shown in this figure, the results of the file transfer can be viewed by using the MQ Explorer.

Transfer request options: Message on command queue

- File transfer request can be created by putting specially formatted message on file transfer agent command queue

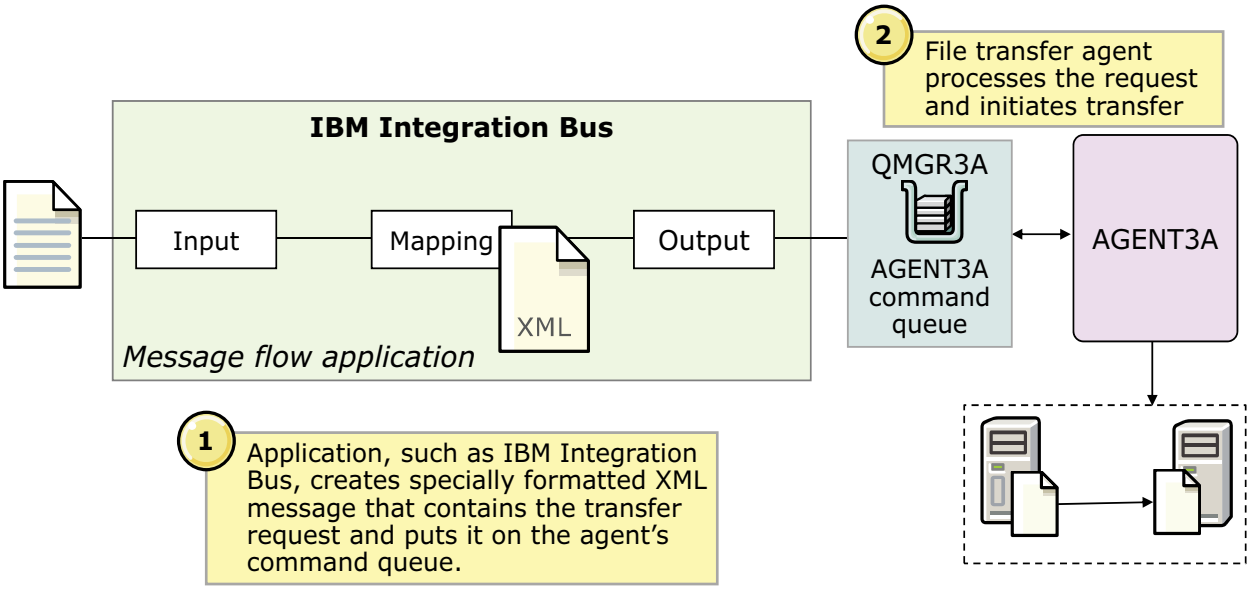


Figure 6-11. Transfer request options: Message on command queue

A third option for requesting a file transfer is putting a specially formatted message on the file transfer agent command queue. If the agent is on a remote server, the message can be placed on a remote queue that is defined with the agent's command queue as a target.

The MQ Managed File Transfer documentation contains details on how to format the message to create a transfer with this method.

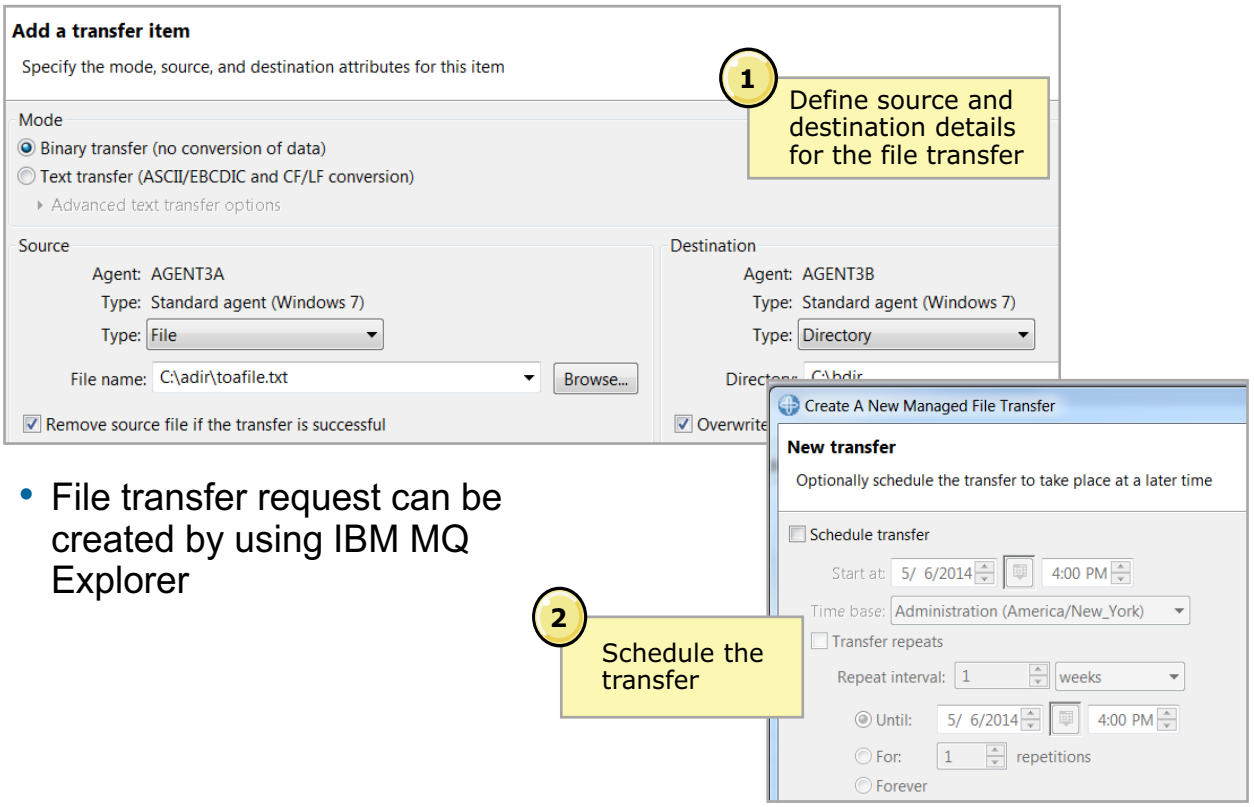
This figure also includes IBM Integration, which can be used to create the file transfer request message. Integration Bus also provides MQ Managed File Transfer input and output message processing nodes. These nodes work with the file transfer agent to transfer the message.

You can put a message on the command queue of a file transfer agent to request that the agent complete one of the following actions:

- Create or cancel a file transfer
- Create or cancel a scheduled file transfer
- Call a command
- Create or delete a monitor
- Return a ping to indicate that the agent is active

IBM Training

Transfer request options: IBM MQ Explorer



- File transfer request can be created by using IBM MQ Explorer

Figure 6-12. Transfer request options: IBM MQ Explorer

Another option for creating a file transfer request is to create the request in MQ Explorer.

With the file transfer options in MQ Explorer, you can create an immediate transfer or a scheduled transfer. You can also specify other actions to take on the source and target files. For example, you can select the **Remove source file if the transfer is successful** check box.

IBM Training

Verifying transfer with IBM MQ Explorer

- View the details of file transfers that are started from a command or MQ Explorer by using the MQ Explorer **Transfer Log**
  - **Source**: Name of the agent on the system where the source file is located
  - **Destination**: Name of the agent on the system you want to transfer the file to
  - **Completion State**: Status of the file transfer
  - **Owner**: User ID on the host that submitted the transfer request
  - **Started:** Time and date that the file transfer agent accepted the transfer request
  - **State Recorded**: Time and date that the completion state was recorded
  - **Job Name**: Identifier that is specified by the user
  - **Transfer ID**: Unique identifier for the file transfer

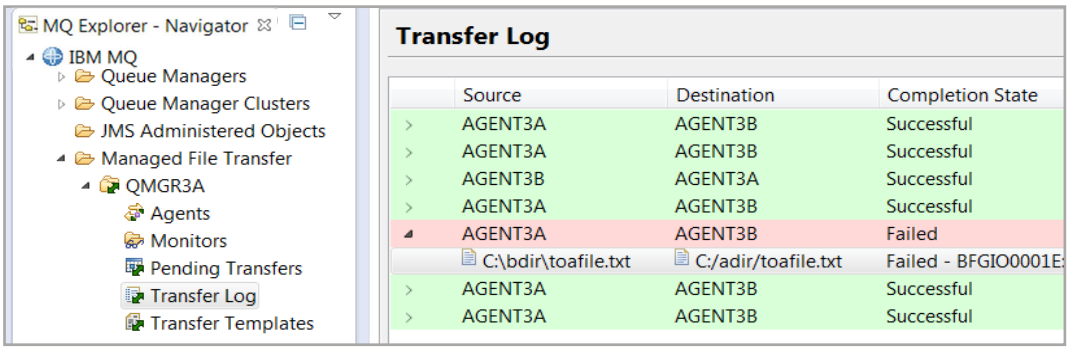


Figure 6-13. Verifying transfer with IBM MQ Explorer

You can use MQ Explorer to verify the results of the transfer by viewing the **Transfer Log**.

You can display the file transfer details by clicking the entry in the **Transfer Log**. The figure summarizes the **Transfer Log** details.

IBM Training

Auditing loggers

- Three types of loggers can be configured to preserve transfer history
  1. Stand-alone file system logger
     - Existing Java process
     - Writes to a local file
     - Not supported on z/OS or IBM i
  2. Stand-alone database logger
     - Requires installation of logger Java application
     - Support DB2 and Oracle databases
     - Requires type JDBC connection to database to use queue manager for global transaction coordination
  3. Java Platform, Enterprise Edition database logger
     - EAR file that is installed in an application server
     - Supports DB2 and Oracle databases
- Loggers are independent from transfer processes

Figure 6-14. Auditing loggers

The status information that is held in MQ Explorer is not kept permanently; however, MQ Managed File Transfer provides three mechanisms to preserve the transfer history.

The stand-alone file logger is a Java process that runs on the system that hosts the coordination queue manager, or on a system that hosts a queue manager with connectivity to the coordination queue manager. The stand-alone file logger is not supported on z/OS or IBM i.

The stand-alone database logger is a Java application that you install on a system that hosts a queue manager and a DB2 or Oracle database. The stand-alone database logger is often installed on the same system as the coordination queue manager. However, it can also be installed on the same system as any queue manager that can connect to the coordination queue manager. The stand-alone database logger uses a type 2 or type 4 JDBC driver to connect to the database. These types of connection are required because the stand-alone database logger uses the queue manager's XA support to coordinate a global transaction over both the queue manager and database, which protects the data.

The Java Platform, Enterprise Edition database logger is provided as an EAR file, which you install into an application server. This logger can be more convenient than using the stand-alone database logger if you have an existing Java Enterprise Edition application server environment. The Java EE database logger is supported for use with DB2 and Oracle databases.

Protocol bridge

- Enables MQ Managed File Transfer (MQMFT) to access files stored on a file server outside the MQ network, either in your local domain or a remote location
- Can use the FTP, FTPS, or SFTP network protocols to exchange files
  - References absolute file paths only
- Each file server requires at least one protocol bridge agent
- Requires file transfer agent to complete file transfers

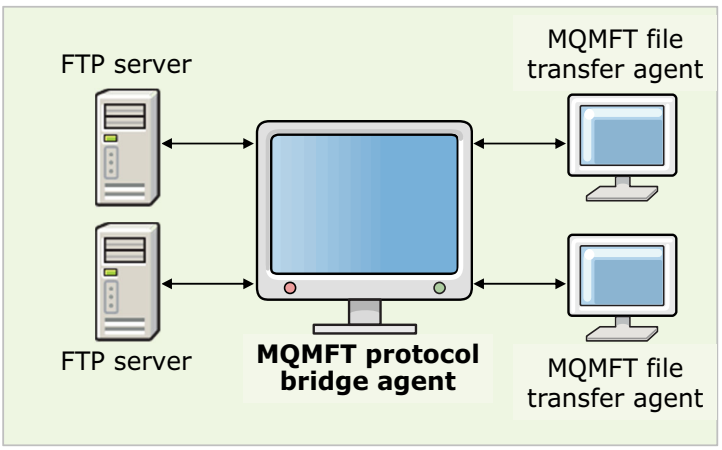


Figure 6-15. Protocol bridge

By default, MQ Managed File Transfer uses the MQ network for the transfer of files. With the MQ Managed File Transfer protocol bridge agent, you can also transfer files over FTP, FTPS, of SFTP.

The protocol bridge agent acts as the source or destination for the file. It does not transfer files, and requires a file transfer agent to transfer of files.

The figure shows two FTP servers, at different locations. The FTP servers exchange files with the MQ Managed File Transfer agents. The protocol bridge agent is between the FTP servers and the rest of the MQ Managed File Transfer network, and is configured to communicate with both FTP servers.

Connect:Direct bridge

- IBM Sterling Connect:Direct
  - Provides security-rich, point-to-point file transfers that can eliminate dependency on unreliable FTP transfers
  - Optimized for high-volume, assured delivery of files within and among enterprises
- Connect:Direct bridge enables file transfers to and from an IBM Sterling Connect:Direct network and MQ Managed File Transfer

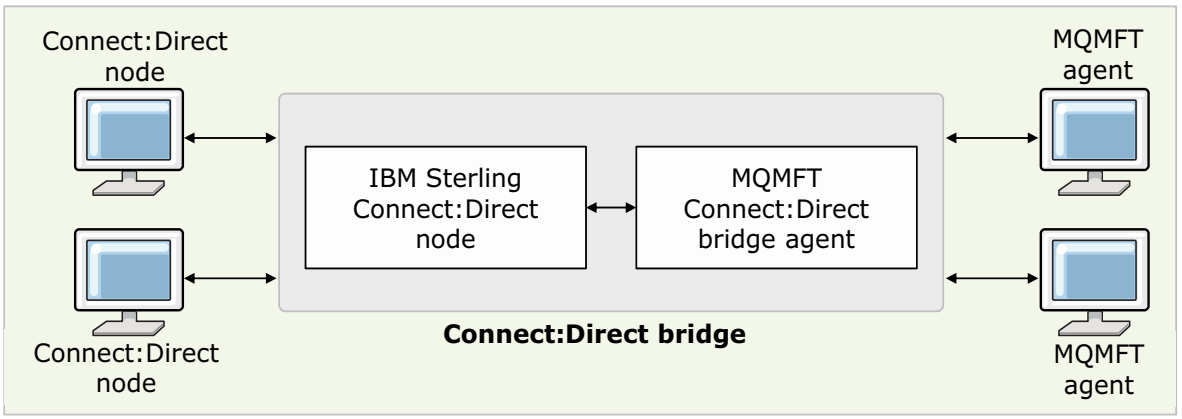


Figure 6-16. Connect:Direct bridge

If you are using IBM Sterling Connect:Direct, you can use the Connect:Direct bridge to transfer files to and from Connect:Direct nodes.

The example in the figure shows an MQ Managed File Transfer (MQMFT) Connect:Direct bridge between two organizations. One organization uses Connect:Direct to transfer files to and from its business partners. The other organization uses MQ Managed File Transfer as its file transfer solution.

By using the MQ Managed File Transfer Connect:Direct bridge, the two organizations can transfer files between the Connect:Direct network and the MQ network.

The Connect:Direct bridge is a component of MQ Managed File Transfer.

IBM Training

Unit summary

- Identify IBM MQ Managed File Transfer scenarios
- Describe the main components of IBM MQ Managed File Transfer
- Describe IBM MQ Managed File Transfer basic configuration
- Identify various ways of initiating a transfer
- Summarize auditing capabilities for IBM MQ Managed File Transfer

Figure 6-17. Unit summary

IBM Training

Review questions

1. The three roles that a queue manager serves in managed file transfer are:
   A. Agent queue manager
   B. Transfer queue manager
   C. Command queue manager
   D. Coordination queue manager

2. True or False: Managed file transfer loggers are independent from the transfer processes and do not need to be running for a transfer to occur.

3. What managed file transfer components enable exchange of files across FTP, SFTP, FTPS, and the managed file transfer network?
   A. Direct:Connect bridge
   B. Coordination queue manager
   C. Protocol bridge
   D. Resource monitors

Figure 6-18. Review questions

Write you answers here:

1.

2.

3.

IBM Training

Review answers

1. The three roles that a queue manager serves in managed file transfer are:
   A. Agent queue manager
   B. Transfer queue manager
   C. Command queue manager
   D. Coordination queue manager

   The answer is A, C, and D.
2. True or False: Managed file transfer loggers are independent from the transfer processes and do not need to be running for a transfer to occur.

   The answer is True.
3. What managed file transfer component enables exchange of files across FTP, SFTP, FTPS, and the managed file transfer network?
   A. Direct:Connect bridge
   B. Coordination queue manager
   C. Protocol bridge
   D. Resource monitors

   The answer is C.

Figure 6-19. Review answers

Unit 7. Introduction to IBM MQ Telemetry and IBM MessageSight

Estimated time

00:30

Overview

This unit describes how IBM MQ Telemetry supports the “Internet of Things” and extends the enterprise to a mobile environment. It also describes how the IBM MessageSight messaging appliance extends the enterprise in a mobile environment with a secure, easy-to-deploy, appliance-based messaging server.

How you will check your progress

- Review questions

References

IBM MQ Library: www.ibm.com/software/integration/wmq/library/index.html

IBM Training

Unit objectives

- Identify the scenarios that require mobile connectivity
- Describe the MQTT protocol
- Describe how IBM MQ Telemetry facilitates connectivity to telemetry devices
- Summarize the main functions of the IBM MessageSight messaging appliance

Figure 7-1. Unit objectives

IBM Training

What is MQTT?

- Machine-to-machine (M2M) or “Internet of Things” (IoT) connectivity protocol
- Lightweight publish/subscribe messaging transport
- For mobile applications where bandwidth and battery power are a premium
- Developed jointly by IBM and Arcom

- Design principles
  - Minimize network bandwidth and resource requirements
  - Attempt to ensure reliability and some degree of assurance of delivery

Figure 7-2. What is MQTT?

MQTT is a simple and lightweight publish/subscribe messaging protocol that is designed for constrained devices and low bandwidth that run on unreliable networks.

The design principles of MQTT are to minimize network bandwidth and device resource requirements while also attempting to ensure reliability and some degree of assurance of delivery. These principles make the protocol ideal for the emerging “machine-to-machine” or “Internet of Things” world of connected devices, and for mobile applications where bandwidth and battery power are at a premium.

Publish and subscribe pattern

- Core elements are clients, connections (sessions), brokers (servers), publishers, subscribers, and topics
  - Messages in MQTT are published to topics (subject areas)
  - Clients register to receive messages by subscribing to a topic

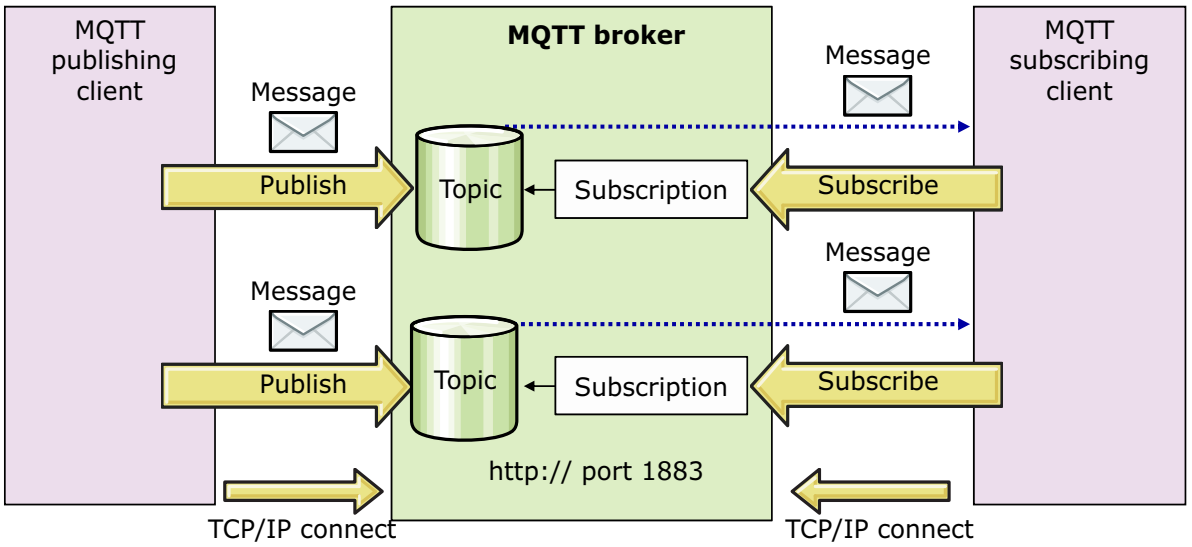


Figure 7-3. Publish and subscribe pattern

The core elements of an MQTT publish/subscribe pattern are clients, brokers, connections between clients and brokers, publishers, subscribers, and topics.

Messages in MQTT are published to topics. Clients register to receive messages by subscribing to a topic. When clients publish messages to a topic, the messages are made available to all subscribers of that topic.

Similar to publish/subscribe in MQ, the MQTT broker manages the subscriptions and topics. The MQTT broker is the server to which message topics are published and from which clients subscribe to topics.

The existence of topics allows the providers and users of information to be decoupled in the publish/subscribe messaging model. This decoupling is achieved by removing the need to include a specific destination in each message as is required in point-to-point messaging.

IBM Training

Some benefits of MQTT

- Open messaging transport that allows MQTT implementations to be created for a wide variety of devices
  - A messaging transport that is independent of the content of the payload
- Clients can run on small footprint devices that have limited resources
- Works efficiently on networks where the bandwidth is low, where cost of sending data is expensive
  - Small transport usage that is designed to minimize network traffic
- Publish/subscribe message pattern provides a one-to-many message distribution
- A mechanism to notify interested parties of an abnormal disconnection of a client
- Supports three qualities of service for message delivery

Figure 7-4. Some benefits of MQTT

This figure lists some of the benefits of MQTT.

MQTT quality of service (QoS)

- MQTT specification defines three qualities of service for message delivery:

QoS value	Description		Occurrence
0	At most once	Fire and forget	<=1
1	At least once	Acknowledged delivery	>=1
2	Exactly once	Assured delivery	=1

- QoS is specified at both publish and subscribe levels
  - A subscriber can downgrade to a lower QoS than is published
  - A subscriber cannot upgrade to a higher QoS than is published

Figure 7-5. MQTT quality of service (QoS)

The MQTT specification defines three qualities of service (QoS) for message delivery:

- QoS 0: The message is either not delivered at all if the client is not connected, or delivered at most one time if the client is connected.
- QoS 1: The message is delivered at least one time. This QoS caters for duplicate messages that are sent until an acknowledged delivery is returned from the client.
- QoS 2: The message is delivered exactly one time. This level is the highest QoS level with assured delivery of the message.

A subscriber can specify a lower QoS, but cannot specify a higher QoS.

MQTT use case: Mobile applications

- MQTT is used in sensors that communicate to an MQTT broker with a satellite link, and in a range of home automation and small device scenarios

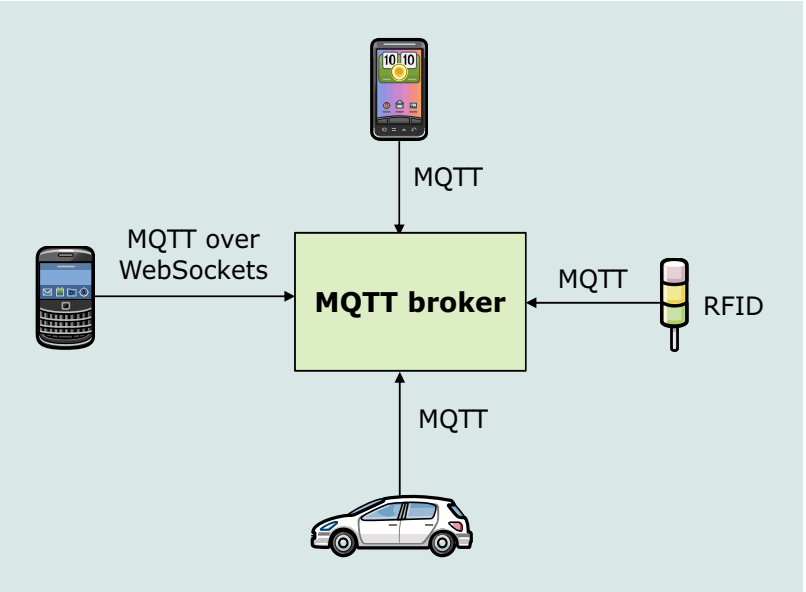


Figure 7-6. MQTT use case: Mobile applications

MQTT is an “Internet of Things” (IoT) connectivity protocol.

A typical use for MQTT is for mobile applications and sensors that communicate to an MQTT broker with a satellite link.

MQTT can work efficiently even on networks with low bandwidth, and high latency, which is the time between the request and response messages.

MQTT broker implementations

- IBM MQ
- IBM MessageSight
- Mosquitto (open source)
- Apache Apollo
- RSMB (Really Small Message Broker): developerWorks
- Eclipse Paho (clients)

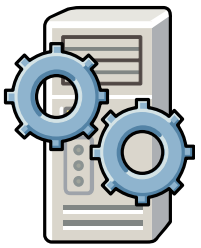


Figure 7-7. MQTT broker implementations

Several open source and vendor MQTT broker implementations are available, including IBM MQ.

IBM Training

IBM MQ Telemetry

- Extends the MQ universal messaging backbone to a wide range of remote sensors, actuators, and telemetry devices
- MQ Telemetry service runs inside the MQ queue manager and uses MQTT protocol to communicate with telemetry devices
- MQ Telemetry channels manage the connection of MQTT clients to MQ
- MQTT daemon for devices
  - Acts as a network concentrator to connect MQTT clients to a single queue manager
  - Provides store-and-forward facilities for small devices that cannot buffer messages during short network outages
- Supports MQ command line and MQ Explorer administrative interfaces
- User-defined MQTT applications control the information that is carried between the telemetry devices and the MQ queue manager, and any actions that are taken in response to that information
  - MQ includes MQTT client libraries to help you create applications
  - Client Software Development Kit is available as the free download

Figure 7-8. IBM MQ Telemetry

IBM MQ Telemetry extends the MQ universal messaging backbone to the “Internet of Things”.

MQ Telemetry includes a telemetry service that is part of a queue manager and command and MQ Explorer administrative interfaces.

MQ Telemetry also includes an MQTT daemon for devices. The daemon is an advanced telemetry client that acts as a network concentrator to connect MQTT clients to a single queue manager. The daemon can also provide store and forward facilities for small devices that cannot buffer messages during short network outages.

MQTT applications that you write control the information that is carried between the telemetry devices and the MQ queue manager. These applications also control any actions that are taken in response to that information. To help create these applications, you use MQTT client libraries.

With MQ Telemetry, you can integrate the collection of data and control of devices with web applications.

IBM Training

IBM MQ Telemetry use case: Smart Health Services

- MQ Telemetry sends secure health data to your hospital and doctor
  - The device requires no user intervention other than power supply, a telephone line, and proximity to the device for part of the day
  - Telemetry channel between the device and the queue manager supports user authentication, communication encryption, and device authentication
  - MQ object authority manager controls access to a publication
- MQTT message alerts or feedback can be sent based on analysis of health data
- Connections are managed by using MQ Explorer

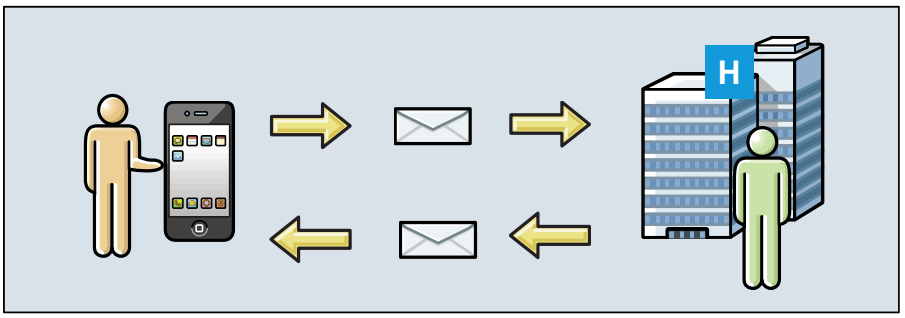


Figure 7-9. IBM MQ Telemetry use case: Smart Health Services

An example of an MQ Telemetry use case is Smart Health Services.

In the collaboration between IBM and a healthcare provider on a cardiac patient care system, an implanted cardioverter defibrillator communicates with a hospital. Data about the patient and the implanted device are transferred by using radio frequency telemetry to the MQTT device in the home of a patient.

Typically, the transfer takes place nightly to a transmitter at the bedside. The transmitter transfers the data securely over the phone system to the hospital, where the data is analyzed.

The system reduces the number of visits a patient must make to a physician. It detects when the patient or device needs attention, and when an emergency occurs, it alerts the on-call physician.

IBM MQ Telemetry use case: Fraud detection

- When a credit card transaction is sent to a bank's server:
  - MQ Telemetry identifies the one person from the thousands and alerts the customer that their card was used
  - MQ Telemetry can use the simplest input of information, and locate the customer

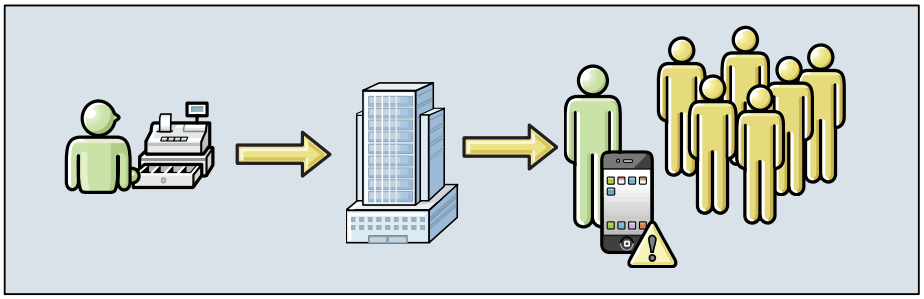


Figure 7-10. IBM MQ Telemetry use case: Fraud detection

In the use case in this figure, MQ Telemetry is used to detect credit card fraud.

Many credit card companies now link a customer's credit card to their mobile phone. When a bank receives a credit card transaction, MQ Telemetry can identify the card owner and send an alert to the card owner's phone that indicates that their card was used.

IBM Training

Benefits of IBM MQ Telemetry

- Message delivery is assured and decoupled from the application
- Application programmers do not need to have communications programming knowledge
- Messages can be exchanged with other messaging applications:
  - Another telemetry application
  - An MQTT daemon for devices
  - An MQI, JMS, or enterprise messaging application
- Message can be exchanged securely by using Transport Layer Security (TLS) and Java Authorization and Authentication Service (JAAS)

Figure 7-11. Benefits of IBM MQ Telemetry

MQ Telemetry adds the benefits of MQ enterprise messaging, such as assured delivery and security, to an MQTT network.

IBM Training

IBM MQ Telemetry security

- TLS can be used to:
  - Encrypt communications between the device and the telemetry channel
  - Authenticate that the device is connecting to the correct server
  - Verify that the client device is authorized to connect to the server
- JAAS can be used to:
  - Check that the user of the device is authorized to use a server application
  - Check with LDAP to verify a password by using a single sign-on directory
- TLS and JAAS can be used together to provide two-factor authentication
  Example: Restrict ciphers that TLS uses to ciphers that meet FIPS standards

Figure 7-12. IBM MQ Telemetry security

MQ Telemetry can add a layer of security to MQTT networks by using Transport Layer Security (TLS) or Java Authentication and Authorization Service (JAAS).

The primary goal of the TLS protocol is to provide privacy and data integrity between two communicating computer applications.

The main goal of JAAS is to separate the concerns of user authentication so that they can be managed independently. JAAS adds a marker about who runs the code, which extends the security architecture for Java applications that require authentication and authorization modules.

It is also possible to use TLS and JASS together to provide two-factor authentication.

IBM Training

IBM MQ Telemetry administration with IBM MQ Explorer

- MQTT subscriptions appear as MQ subscriptions and can be managed from **Subscriptions** view
- MQTT and MQ share the topic space
  - Can publish from MQTT to MQ and from MQ to MQTT
- MQ Explorer provides a sample MQTT configuration and MQTT client utility

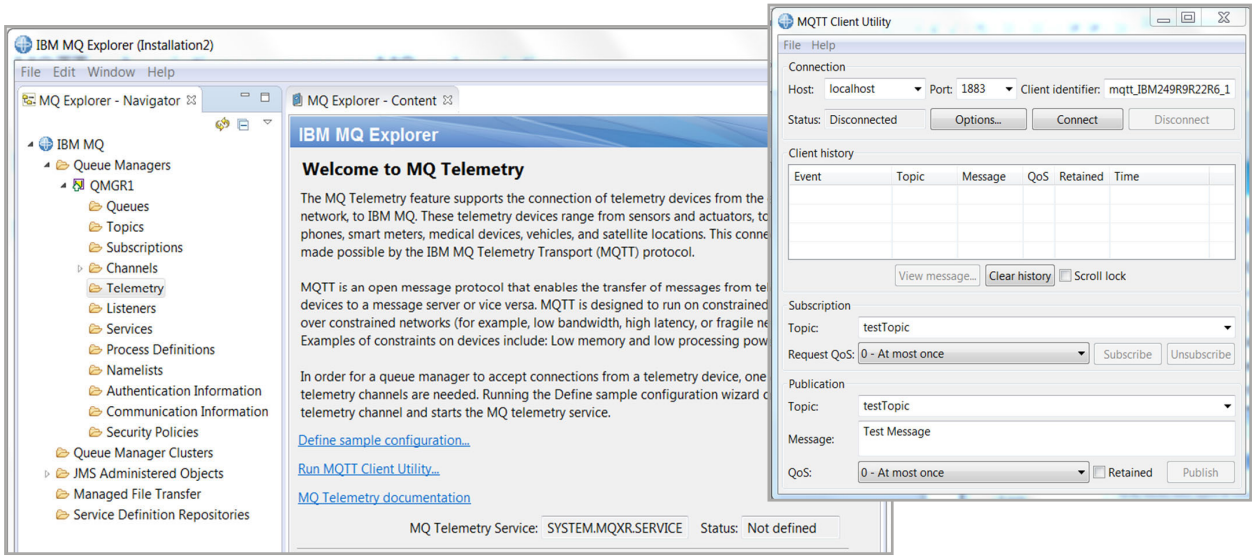


Figure 7-13. IBM MQ Telemetry administration with IBM MQ Explorer

When you install MQ Telemetry on Windows or Linux, it adds administrative functions for MQTT to MQ Explorer.

You can manage MQTT topics and subscriptions by using the same interface that you use to manage MQ topics and subscriptions. For example, both the MQ and MQTT subscriptions are listed in the MQ Explorer **Subscriptions** view.

You can use the MQTT client utility to validate a basic or custom MQ Telemetry configuration by connecting, subscribing to topics, and publishing messages. The MQTT client utility also provides a simple tool to aid in debugging MQ Telemetry applications.

IBM Training

IBM MQ Telemetry sample configuration

- Defines and starts the MQ Telemetry service
- Defines and sets an MQ Telemetry transmission queue on the queue manager
- Enables MQ applications to send point-to-point messages to MQTT clients without creating a separate queue manager alias for every client
  - Messages that are destined for MQTT clients are routed through the queue manager's MQTT transmission queue to the client whose client identifier matches the queue manager name that the message is sent to
- Allows messages to be sent to clients attached to the MQTT Listener
- Supports publications and subscriptions on any topic
- Defines a sample MQ Telemetry channel

Figure 7-14. IBM MQ Telemetry sample configuration

The MQ Telemetry page in MQ Explorer includes a link to create a sample configuration. You can use the sample configuration and the MQTT client utility in MQ Explorer to verify that the MQ Telemetry components are installed. You can also check that publish/subscribe works correctly.

The sample configuration defines and starts the MQ Telemetry service, defines and sets an MQ Telemetry transmission queue on the queue manager, and defines a sample MQ Telemetry channel.

What is IBM MessageSight?

- An edge-of-network appliance for handling high volumes of messages that are sent from a range of sensors and mobile devices
- A messaging server able to support clients that are connected to it using various protocols
  - MQTT over TCP/IP
  - MQTT over WebSockets
  - JMS
- IBM offers a virtual IBM MessageSight image for development

Figure 7-15. What is IBM MessageSight?

IBM MessageSight is an appliance-based messaging server that is designed to handle large numbers of connected clients and devices and process high volumes of messages with consistent latency.

MessageSight is a messaging server, able to support clients that are connected to it using various protocols.

The messaging and communication standards that MessageSight uses includes MQTT.

IBM offers developers a virtual MessageSight image that can be downloaded at no charge to develop mobile and M2M applications for production use with a physical MessageSight appliance.

IBM Training

Edge of network messaging appliance

- IBM MessageSight is a highly scalable middleware messaging product that provides the full-duplex web communication that is required for rich Internet, intranet, and mobile applications

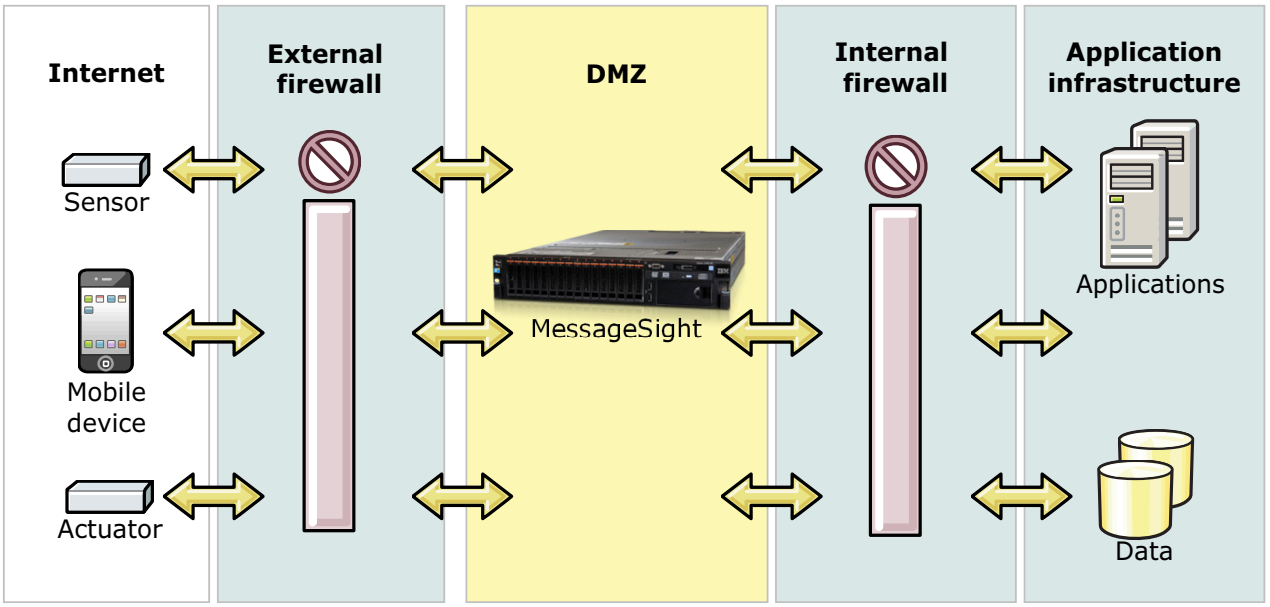


Figure 7-16. Edge of network messaging appliance

MessageSight is a highly scalable middleware messaging product that provides the full-duplex web communication that is required for rich Internet, intranet, and mobile applications.

Many devices such as sensors, monitors, machinery, cars, and mobile devices can publish data, read data, or both. On the other side of the network, applications can read data from these devices, send data, or both.MessageSight acts as the messaging gateway for all of these connected devices.

MessageSight is a topic-based, publish/subscribe broker, not a content-based broker. Unlike a content-based broker, MessageSight does not query or alter the message-content.

No user-based applications can run on MessageSight.

IBM Training

IBM MessageSight appliance capabilities

- Extends the IBM Messaging family with a secure, easy-to-deploy, appliance-based messaging server
  - Optimized to address the massive scale requirements of machine-to-machine and mobile use cases
- Supports 1 million connections and millions of messages per second
- Uses hardware acceleration for performance
- Designed to sit at the edge of the enterprise
- Can be used to extend an existing messaging infrastructure or used as a stand-alone product
- Complements IBM MQ
  - Provides an offload and accelerator for edge of enterprise scenarios
- Supports familiar APIs with a mixture of standard and high-speed protocols

Figure 7-17. IBM MessageSight appliance capabilities

MessageSight extends the IBM messaging family with a secure, easy-to-deploy appliance-based messaging server, optimized to address the massive scale requirements of “machine-to-machine” and “Internet of Things” use cases.

The messaging appliance supports up to 1 million concurrent connections and ultra high volumes of messages per second. It uses hardware acceleration for enhanced performance.

MessageSight is deployed at the edge of the enterprise. It can be used to extend an existing messaging infrastructure or used as a stand-alone product.

MessageSight supports APIs with a mixture of standard and high-speed protocols.

IBM Training

MessageSight architecture overview

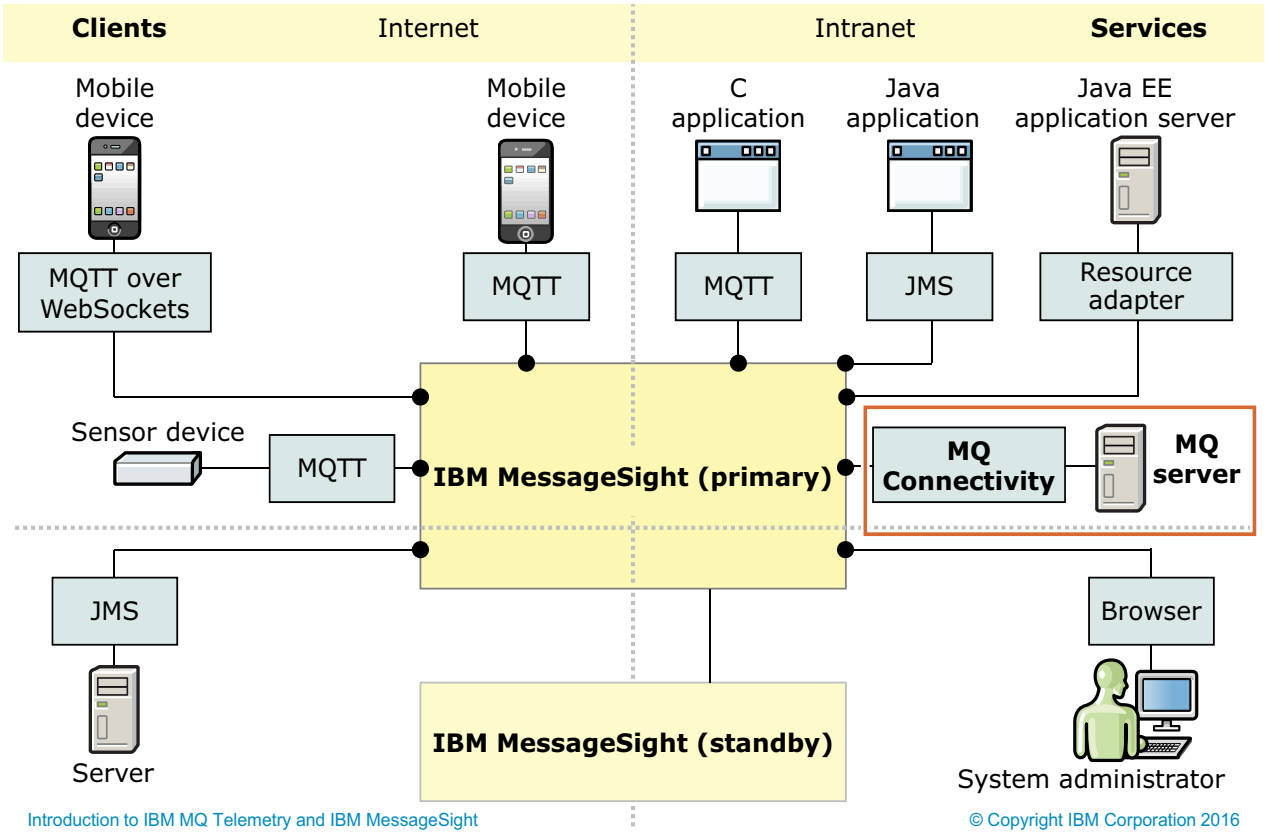


Figure 7-18. MessageSight architecture overview

The MQ Connectivity feature of MessageSight enables messages from MessageSight to be forwarded to MQ, or for messages from MQ to be forwarded to MessageSight.

To use the MQ Connectivity feature, you must configure two components:

- Configure the connections to one or more MQ queue managers.
- Define the destination mappings rules that govern which messages are forwarded to and from the queue managers.

The MQ Connectivity options can be configured with the general administrative tasks from MessageSight or the command interface.

IBM Training

IBM MessageSight example use cases

- Connected city
  - Traffic monitoring and alerts, emergency detection and response
  - Crowd-sourced monitoring of traffic and notification to drivers for rerouting and assistance
- Connected car
  - Vehicle telematics for security and customer service
  - Notification of location and state
- Process and utility industries
  - Remote monitoring and control
  - Analysis and prediction of maintenance
- Mobile applications
  - Secure, reliable, and fast messaging
  - Personal investment portfolio and market news

Figure 7-19. IBM MessageSight example use cases

This figure lists some example use cases for MessageSight.

- A connected city can use messages for traffic monitoring and alerts that aid with emergency detection and response. Messages that are received from crowd-sourcing reporting of traffic conditions can be used to notify other drivers of accidents and suggest alternative directions.
- Connected cars use sensors that are in the cars for security and customer service. These vehicle telematics can be used for vehicle tracking and location and to signal and schedule preventive maintenance.
- Process and utility industries can use automated messaging for the remote monitoring and control of devices. Utility companies can remotely read electricity meters. Pipeline operators can monitor the flow of liquids or gases in pipelines and remotely turn valves on or off.
- Mobile applications can use telematics for secure, reliable, and fast messaging.

IBM Training

IBM MessageSight: Secure and reliable

- Hardened form factor
  - Secure firmware that is signed and encrypted by IBM
  - No user-visible, general operating system
- Security
  - Support for TLS, SSL security
  - Authentication with user ID and password, or by certificate
  - Easy to configure security profiles where each endpoint (port) can be configured
- High availability
  - Appliances can be paired for failover support
  - The primary appliance is the one that is processing messages
  - The standby appliance is the one to which the primary appliance is replicated
  - The primary node continually replicates both the message store and appliance configuration information to the standby node

Figure 7-20. IBM MessageSight: Secure and reliable

For a more secure administration environment, configure MessageSight to ensure that the management interface is not accessible through the Internet.

Transport level security is controlled by the security profile that is associated with an endpoint. A security profile defines the security operations that are applied to a message flow. Each security profile has an associated certificate profile that defines the server certificate to use. Server certificates protect against impersonation, certifying that a public key belongs to a specified entity.

Client authentication occurs when MessageSight requests a certificate from a client to verify that the client is who it claims to be. When a messaging client connects to MessageSight, you can choose whether you want to authenticate the messaging user ID and password of that client.

The high availability nature of MessageSight is its ability to withstand software or hardware outages and continue providing processing capability. For MessageSight, a pair of appliances is configured for high availability. The primary appliance is the one that is processing messages. The standby appliance is the device to which the primary appliance is replicated.

After the primary and standby nodes are physically connected and configured, the primary node continually replicates both the message store and appliance configuration information to the standby node. If the primary node fails, the standby node has the most recent data that is needed for applications to continue messaging services.

IBM Training

IBM MessageSight web user interface

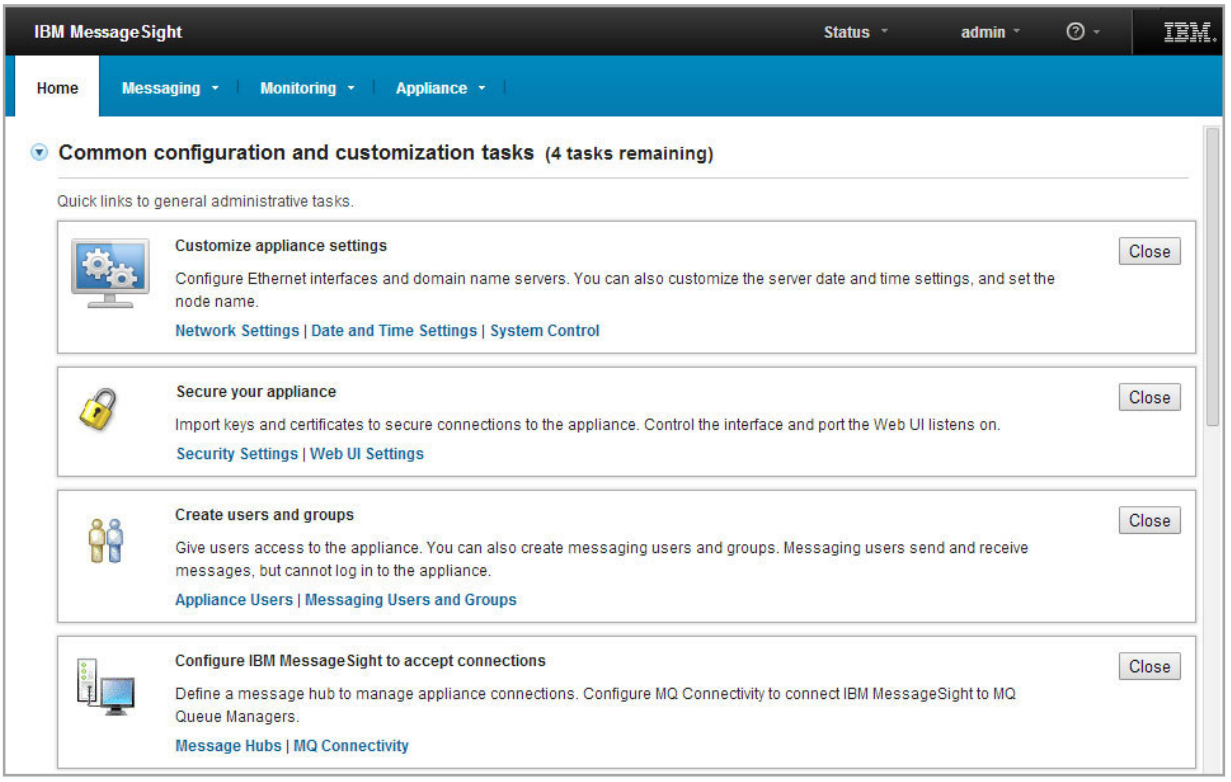


Figure 7-21. IBM MessageSight web user interface

You can use the MessageSight web user interface or the command interface to administer your MessageSight environment.

This figure shows the MessageSight web user interface **Home** tab.

IBM Training

Unit summary

- Identify the scenarios that require mobile connectivity
- Describe the MQTT protocol
- Describe how IBM MQ Telemetry facilitates connectivity to telemetry devices
- Summarize the main functions of the IBM MessageSight messaging appliance

Figure 7-22. Unit summary

IBM Training

Review questions

1. What descriptions are true about MQTT?
   A. A machine-to-machine connectivity protocol
   B. Lightweight publish and subscribe messaging transport
   C. Created for mobile applications where bandwidth and battery power are a premium
   D. Supports four quality of service levels

2. True or False: MQTT supports one standard class of service.

3. True or False: MQTT is a transport, and IBM MessageSight is an appliance that sits at the edge of the network to handle high volumes of messages.

Figure 7-23. Review questions

Write your answers here:

1.

2.

3.

IBM Training

Review answers

1. What descriptions are true about MQTT?
   A. A machine-to-machine connectivity protocol
   B. Lightweight publish and subscribe messaging transport
   C. Created for mobile applications where bandwidth and battery power are a premium
   D. Supports four quality of service levels

   The answer is A, B, and C.

2. True or False: MQTT supports one standard class of service.
   The answer is False. MQTT supports three levels of service.

3. True or False: MQTT is a transport, and IBM MessageSight is an appliance that sits at the edge of the network to handle high volumes of messages.
   The answer is True.

Figure 7-24. Review answers

Unit 8. Introduction to the IBM MQ Appliance

Estimated time

00:30

Overview

This unit summarizes the features of the IBM MQ Appliance messaging server.

How you will check your progress

- Review questions

References

IBM MQ Library: www.ibm.com/software/integration/wmq/library/index.html

IBM Training

Unit objectives

- Summarize the main functions and benefits of the IBM MQ Appliance
- Compare the IBM MQ Appliance with IBM MessageSight
- Compare the IBM MQ Appliance with IBM MQ

Figure 8-1. Unit objectives

IBM Training

New challenges for enterprise messaging

- How do you deploy to a location where no local product knowledge is available?
- How do you keep systems updated painlessly?
- How do you ensure security when working with partners?
- How do you reduce the number of servers?
- How do you work with a mixture of hardware, operating systems, and versions?
- How do you support failover?
- How do you maintain business standards when adding partners?

Figure 8-2. New challenges for enterprise messaging

After implementing MQ, its use typically grows. Growth brings new challenges, such as an increase the number of servers that are managed, and keeping those servers up to date, including hardware, operating system. For MQ, challenges include deploying MQ in locations where no MQ knowledge exists, ensuring security when establishing connections to partners, and supporting 24 x 7 operations.

In this unit, learn how the MQ Appliance can help solve your enterprise messaging challenges.

IBM Training

Introducing IBM MQ Appliance

- Provides scalability and security of MQ in the convenience, fast time-to-value, and low total cost of ownership of an appliance
  - Integrates seamlessly into MQ networks and clusters
  - Familiar administration model for administrators with MQ skills
  - Supports MQ Light API
- Ideal for use as a messaging hub that runs queue managers that are accessed by clients, or to extend MQ connectivity to a remote location
- Familiar feel for existing 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 your applications to the cloud as part of your hybrid infrastructure

Introduction to the IBM MQ Appliance

Figure 8-3. Introducing IBM MQ Appliance

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

Why an appliance?

- Fixed hardware specification allows IBM to simplify and tune the firmware
  - Fewer “points of variability” make it easier to deploy and manage
  - Less performance tuning required
- Standardization accelerates deployment
  - Repeatable and fast, less configuration and tuning required
  - Post-deployment resource definition or lockdown before deployment
- “Hub” pattern separates messaging from applications and middleware
  - Organizational independence from application teams
  - Improved availability due to reduction of downtime
  - Predictable performance for simpler capacity planning
- Simplified ownership
  - Self-contained: Avoids dependencies on other resources and teams
  - Licensing: Simpler than calculating licensing costs
  - Security: Easier to assess for security compliance audit

Figure 8-4. Why an appliance?

The MQ Appliance is a hardware and firmware platform, so no user operating system exists in the traditional sense. The MQ firmware components are a modified build of the most recent version of MQ for distributed operating systems.

The state-of-the-art hardware allows IBM to simplify and tune the firmware. It means fewer points of variability for your MQ servers, each with its operating systems and versions.

IBM offers two models of the MQ Appliance to fit different uses and performance requirements. The MQ Appliance also offers you the ability to lock down your configuration before deployment of the appliance to a location that lacks an MQ administrator.

The MQ Appliance hub pattern separates messaging from the applications, so the appliance is your messaging hub only. You connect all of your MQ client applications, which provides more predictable performance and makes planning for capacity simpler. The built-in high availability improves availability with a reduced downtime.

The MQ Appliance is self-contained, which means fewer dependencies on other resources and teams exist.

IBM Training

IBM MQ Appliance M2001

- Simple application that can be rapidly integrated for high-availability and synchronous connectivity to a paired appliance
- Built-in disaster recovery for recovery over longer distances
- MQ Console provides a browser-based user interface, offering personalized monitoring and configuration
- Simple maintenance with FixPacks that are delivered as certified firmware updates onto a locked down appliance
- Appliance delivered pre-optimized for maximum performance
- Includes IBM MQ Advanced Message Security

Figure 8-5. IBM MQ Appliance M2001

The IBM MQ Appliance is available in two options:

- IBM MQ Appliance M2001A for larger enterprise workloads.
- IBM MQ Appliance M2001B for smaller workloads and lower processing capability at a lower price.

The MQ Appliance keeps data safe. High availability automatically recovers data locally and synchronously through pairing with a second appliance. Data recovery recovers data over large distances asynchronously to one or more appliances that are deployed remotely.

The MQ Console is a browser-based user interface for configuring and monitoring the MQ Appliance. The MQ Console can be personalized and customized for each user. The MQ Console works with the existing MQ administration interfaces such as MQ Explorer and MQSC.

IBM Training

New features in the IBM MQ Appliance M2001

- Higher, faster message throughput
- Improved performance
  - Change to two high capacity (3.2 TB) solid-state drives
  - Increase from two to four 10 GB Ethernet ports
- Better data protection with built-in security
- Extension to the Cloud with support of the MQ Light API

Figure 8-6. New features in the IBM MQ Appliance M2001

The MQ Appliance M2001 includes 192 GB of memory, a serial port, two 1 Mb Ethernet ports for management, eight 1 Mb Ethernet ports, four 10 Gb Ethernet ports, and two 3.2 TB solid-state drives.

The MQ Appliance M2001 also includes support for the MQ Light API and full support for disaster recovery.

IBM Training

IBM MQ Appliance administration options

- Command interface
  - Supports appliance-specific commands such as configuring network interfaces and importing certificates
  - Offers a familiar subset of MQ control commands
  - Use MQSC interactively, or run MQSC scripts remotely
- MQ Console
  - Browser-based user interface
  - Avoids maintenance of “rich client” installations
- MQ Explorer
  - Essential for existing MQ administrators
- PCF
  - Supports remote administration by using all of the existing MQ tools

Figure 8-7. IBM MQ Appliance administration options

With the MQ Appliance, you have a choice of administration interfaces.

First, the MQ Appliance contains some appliance-specific commands to configure and control the appliance hardware, such as for the network interfaces.

The MQ Appliance also supports a subset of MQ control commands for creating and managing queue managers. You can run MQSC interactively on the MQ Appliance, or run MQSC scripts remotely, connecting as a client.

You can configure queue managers that are created on the MQ Appliance as remote queue managers by using the MQ Explorer.

The MQ Appliance also supports PCF, which enables remote administration and monitoring from existing MQ tools.

The MQ Console, a browser-based user interface for administering the MQ Appliance.

IBM MQ Console

- Browser-based administration console for managing the MQ Appliance and MQ objects
- View, create, and delete:
  - Queue managers
  - Queues and topics
  - Channels and channel authentication records
  - Listeners
- User-definable chart widgets support monitoring queue manager and the MQ Appliance
- Dashboard can be customized for each user

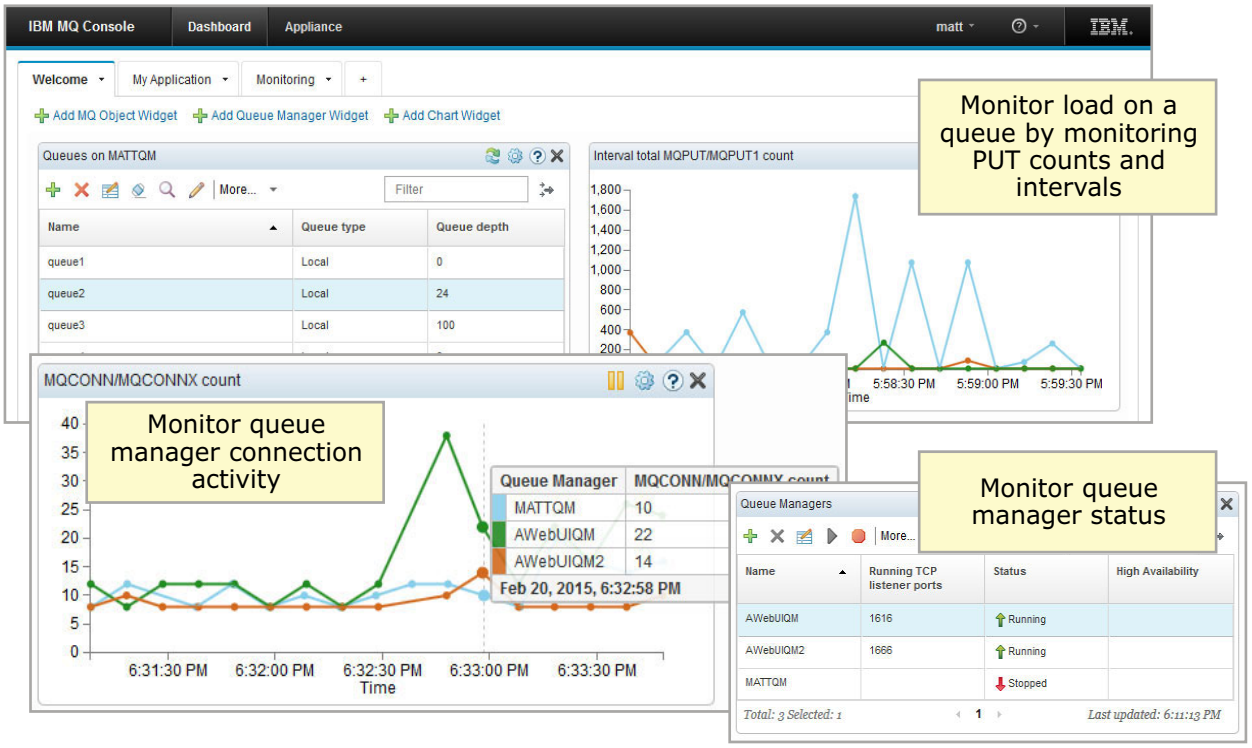
Figure 8-8. IBM MQ Console

The MQ Console eliminates the need to install and maintain client installations. It allows the administration of MQ and offers monitoring capabilities.

The MQ Console includes user-definable chart widgets and dashboard so that each user can customize the interface and monitoring reports to fit their requirements.

IBM Training

IBM MQ Console monitoring examples



See the demonstration of the IBM MQ Console for the IBM MQ Appliance on YouTube: https://ibm.biz/BdXcq3

Figure 8-9. IBM MQ Console monitoring examples

In the MQ Console, you can add widgets to show and administer MQ objects. You can also add chart widgets to show monitoring statistics, for the MQ Appliance and for MQ.

You can create as many pages as you want, and lay out widgets how you want. Your configuration customization is stored; if you log in from multiple locations or devices, you always get your custom interface.

IBM Training

Connectivity

- IBM MQ Appliance supports a number of protocols for message transmission
  - MQ client protocol for connectivity from applications
  - MQ server protocol for connectivity with other MQ queue managers and MQ Explorer
  - AMQP channel for MQ Light API client connectivity

Figure 8-10. Connectivity

The MQ Appliance supports many protocols for message transmission between MQ queue managers, client applications, and MQ Explorer.

The MQ Appliance also supports an AMQP channel for MQ Light API client connectivity.

IBM Training

Is message content secure with the IBM MQ Appliance?

IBM MQ AMS is built in at no additional cost, which provides:

- End-to-end security
  - Protects data at rest in queues; detects and removes rogue messages
  - Authenticates and protects messages across the enterprise
  - Applies end-to-end encryption to existing systems with minimal disruption
- Administrative logging
  - Reduces the scope and costs of audits
  - Proves that data is not captured in logs, memory dumps, and traces
  - Provides separation of duties for administrators

Figure 8-11. Is message content secure with the IBM MQ Appliance?

Included in the MQ Appliance is MQ Advanced Message Security, which allows for end-to-end encryption of message contents at rest, based on pre-defined policies.

MQ Advanced Message Security ensures that businesses that need to encrypt message contents can include MQ Appliances in their environments. Messages that are stored in the MQ appliances are encrypted and their contents are protected.

IBM Training

More about security

- Appliance administrator can be authorized to administer MQ
  - Can separate roles of appliance administrator, messaging administrator, and messaging users
  - Messaging users can send and receive messages, and can be authorized to remotely manage some aspects of queue managers by using tools such as IBM MQ Explorer
- Secure connectivity over SSL/TLS
- Scalable security administration
  - For few messaging users, you can define them locally
  - For larger communities, you can use an external repository such an LDAP repository
- Appliance contains pre-optimized security features that cannot be altered

Figure 8-12. More about security

The MQ Appliance supports multiple user roles for administration.

For example, an MQ Appliance administrator administers the MQ Appliance hardware itself. This person can be authorized to administer MQ too, or the roles can be separated between the MQ Appliance administrator and the MQ administrator. Both of these roles are separate from messaging users, who are the users that send and receive messages. You can authorize a set of messaging users to manage some aspects of queue managers remotely, by using tools such as the MQ Explorer.

You can define messaging users on the MQ Appliance. For larger communities, you can connect to an external LDAP repository. The advantage of an LDAP repository is that you define user once in the repository, not directly on multiple MQ Appliances.

The MQ Appliance supports connectivity over SSL/TLS. You can import certificates in the MQ Appliance or for testing, you can create self-signed certificates. With MQ Advanced Message Security, you can define profiles to apply to queues to keep messages that are digitally signed so they cannot be tampered with, or to also be encrypted.

IBM Training

Disaster recovery planning

- Back up MQ Appliance configuration, and restore to the same or a different MQ Appliance with the same firmware levels
- Also back up:
  - Appliance messaging user accounts
  - Queue manager key repository
  - Queue manager configuration
  - Configuration data for MQ Console users
  - Message data

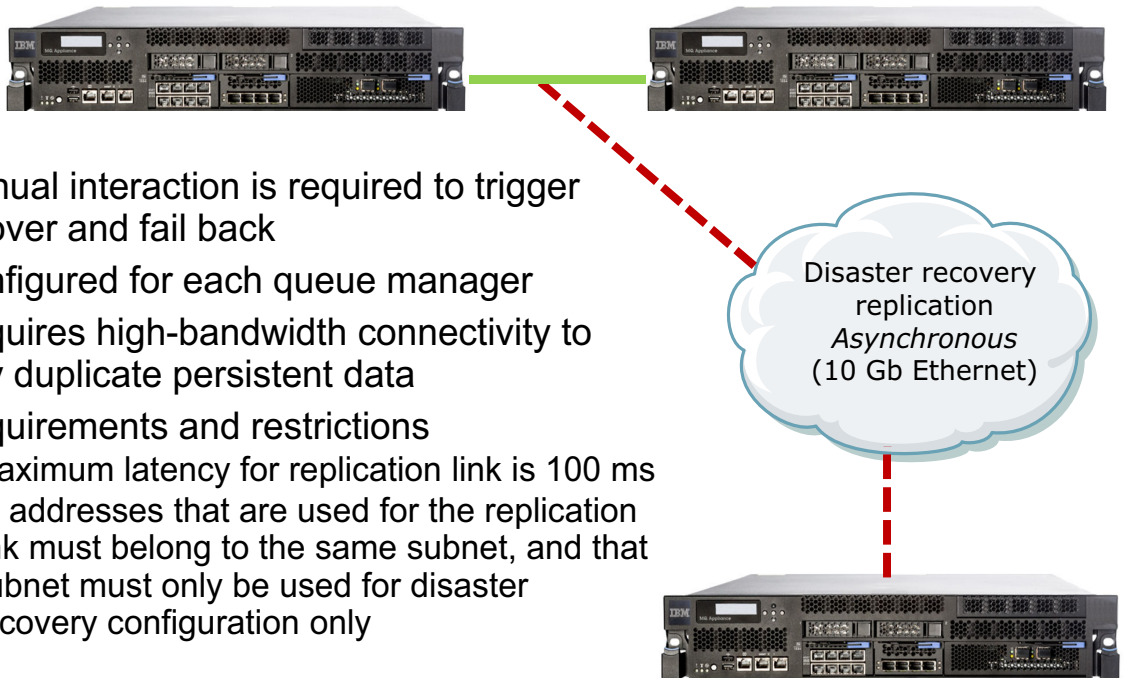
Figure 8-13. Disaster recovery planning

The MQ Appliance supports commands to back up your MQ Appliance configuration, and restore it to the same or a different appliance.

You can do a full or partial back-up. Partial back-up options include the MQ Appliance messaging user accounts, the key repository, queue manager configurations, MQ Console user configurations, and message data.

IBM Training

Disaster recovery



- Manual interaction is required to trigger failover and fail back
- Configured for each queue manager
- Requires high-bandwidth connectivity to fully duplicate persistent data
- Requirements and restrictions
  - Maximum latency for replication link is 100 ms
  - IP addresses that are used for the replication link must belong to the same subnet, and that subnet must only be used for disaster recovery configuration only

Figure 8-14. Disaster recovery

The MQ Appliance supports disaster recovery. At least two MQ Appliances are required to use the disaster recovery support. The example that is shown in the figure uses three MQ Appliances.

In disaster recovery, one MQ Appliance is designated as the *primary instance* and the second is the *secondary instance*.

Data is usually sent from the primary instance of a queue manager to the secondary instance of the queue manager. Data can be sent in two modes: Replication mode and Synchronization mode.

- In Replication mode, the updates to the storage of the primary instance of the queue manager are sent asynchronously to the secondary instance of the queue manager, in the order in which they are made. If the primary instance fails, the secondary instance can be started.
- Synchronization mode is used if the network connection between the two instances is lost. In this mode, the primary instance stops trying to send updates to the secondary instance. When the network connection is restored, the primary instance sends the smallest amount of information necessary to get the secondary instance to the same state as the primary instance as quickly as possible.

IBM Training

What is the value to you?

- Integrated high availability and disaster recovery
- Easy to secure
  - Physically tamper-proof
  - IBM MQ Advanced Message Security provides data encryption
- Pre-optimized, quality hardware
  - No tuning necessary
  - Includes large capacity SSDs
  - 10 GB network speed with 4 x 10 GB Ethernet ports
- Simple administration with browser-based MQ Console
- Simplified audits for security compliance requirements
- New opportunities through integration with cloud
  - New data endpoints
  - Increased agility
- Simple migration of an appliance to new versions and updates
  - Certified hardware from IBM
  - Single firmware updates are quick and easy to apply
- A dedicated appliance
  - Less diverse hardware and operating systems of MQ deployments, easier to manage, quicker to deploy
  - Lower total cost of ownership through consolidation of MQ
- Multiple queue managers supported on a single appliance
  - Reduces data center space, power costs, and management burden
- Flexibility and rapid deployment
  - Supports existing applications that are built to use MQ client connections

Figure 8-15. What is the value to you?

The MQ Appliance is a dedicated appliance. It is quick to deploy and offers the ability to consolidate multiple MQ queue managers onto a single appliance. An appliance lowers the total cost of ownership and can help reduce the number of servers that use MQ.

The MQ Appliance is a tamper proof appliance. It includes MQ Advanced Message Security, which provides built-in end-to-end data encryption of messages in motion and at rest.

The MQ Appliance is pre-tuned and optimized to provide the best performance without tuning.

The MQ Appliance supports traditional MQ administration interfaces and the MQ Console.

With the MQ Appliance, maintenance becomes simpler because you apply a single firmware image that is quick and easy to apply.

The MQ Appliance supports built-in high availability by pairing two appliances for greatly simplified and reliable failover. It also supports flexible built-in disaster recovery.

Comparison between IBM messaging appliances

IBM MessageSight

- MQTT and JMS connections to support edge, mobile, and machine-to-machine (M2M) and Internet of Things (IoT) device messaging
- For deployment in the DMZ or behind the firewall
- Virtual appliance available for development

IBM MQ Appliance

- MQ connectivity to support enterprise messaging
- For deployment behind the enterprise firewall
- Physical appliance only

Figure 8-16. Comparison between IBM messaging appliances

Although IBM MessageSight and the IBM MQ Appliance are both messaging appliances, significant differences exist between them.

MessageSight is designed as an edge of network secure messaging concentrator that supports MQTT and JMS. It does not contain an MQ server or queue managers. It can connect to MQ on a server or in an MQ Appliance as a client.

IBM Training

Comparing IBM MQ to the IBM MQ Appliance

IBM MQ		IBM MQ Appliance
Configured by users. Requires network-attached storage.	**High availability**	Ready to use high Availability, by using paired appliances
Asynchronous replication of network-attached storage for off-site backup	**Disaster recovery**	Flexible asynchronous replication between appliances across longer distances than HA
Deploying MQ on new hardware and operating systems requires specific skills and knowledge	**Deployment time and complexity**	Deploy messaging hubs centrally or remotely in minutes out of the box
MQ environment might be spread over hundreds of different, diverse servers	**MQ network footprint**	Less hardware is required, so less maintenance, easy to update, less variability
MQ Explorer and MQSC	**Administration tools**	MQ Explorer, MQSC plus customizable MQ Console
Administrators manually tune MQ to optimize performance	**Performance tuning**	Pre-optimized for best performance without tuning
Developers can customize by using user exits	**Customization and user exits**	No user exits or local application code

Figure 8-17. Comparing IBM MQ to the IBM MQ Appliance

This figure summarizes some of the differences between running MQ on a server and running MQ on the MQ Appliance. The biggest difference is the optimization that an appliance provides.

IBM Training

Can you use IBM MQ and the IBM MQ Appliance together?

YES!

- MQ Appliance works seamlessly alongside other MQ or MQ Advanced deployments
- No changes are required for MQ clients
- MQ clusters can be a mix of MQ servers and MQ Appliances
- MQ Advanced deployments can be on the same server as applications and connect to MQ Appliance hubs

Figure 8-18. Can you use IBM MQ and the IBM MQ Appliance together?

Enterprise messaging can include both MQ server installations and MQ Appliances. For example, you can have clusters of MQ queue managers that span MQ Appliances and MQ installed on server systems.

IBM Training

Unit summary

- Summarize the main functions and benefits of the IBM MQ Appliance
- Compare the IBM MQ Appliance with IBM MessageSight
- Compare the IBM MQ Appliance with IBM MQ

Figure 8-19. Unit summary

IBM Training

Review questions

1. True or False: An IBM MQ Appliance cannot communicate with MQ queue managers that run an AIX server.

2. True of False: The IBM MQ Appliance M2001 includes IBM MQ Advanced Message Security.

Figure 8-20. Review questions

Write your answers here:

1.

2.

Review answers

1. True or False: An IBM MQ Appliance cannot communicate with MQ queue managers that run an AIX server.
   The answer is False. An MQ network can include both IBM MQ Appliances and MQ server installations on supported operating system servers.

2. True or False: The IBM MQ Appliance M2001 includes IBM MQ Advanced Message Security.
   The answer is True.

Figure 8-21. Review answers

Unit 9. Expanding the scope of IBM MQ

Estimated time

00:45

Overview

This unit provides a brief overview of various linking and bridging mechanisms that are available to expand the scope of messaging and queuing in an enterprise to other IBM products.

How you will check your progress

- Review questions

References

IBM MQ Library: http://www.ibm.com/software/integration/wmq/library/index.html

IBM Training

Unit objectives

- Identify the IBM MQ z/OS links and bridges
- Describe how IBM MQ can be used as part of the communications infrastructure to act as a JMS provider
- Describe how IBM MQ interfaces with IBM Integration Bus
- Distinguish ways that IBM MQ is used by WebSphere Application Server
- Describe the options for deploying IBM MQ in a Cloud environment

Figure 9-1. Unit objectives

IBM Training

Using IBM MQ with IMS

- IBM MQ IMS adapter
  - Interface between IMS application programs and an MQ subsystem
  - Different application environments can send and receive messages through a message queuing network
  - IMS application programs can use the MQI
- IBM MQ IMS bridge (optional)
  - IMS Open Transaction Manager Access (OTMA) client
  - Enables implicit MQI support
  - Use existing applications with MQ, without rewriting them

Figure 9-2. Using IBM MQ with IMS

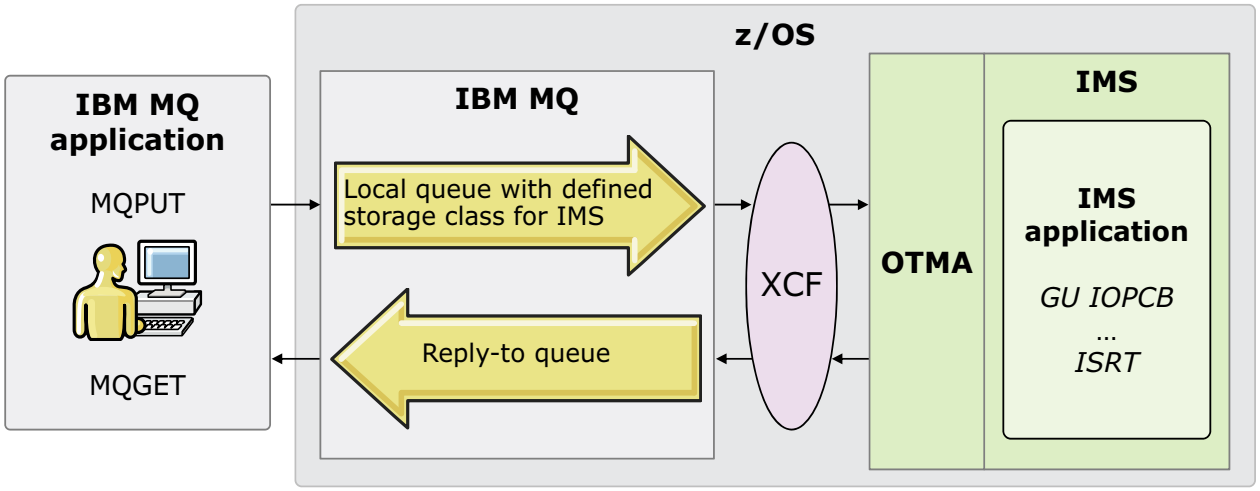
The IBM MQ IMS adapter and the IBM MQ IMS bridge are the two components that allow MQ to interact with IMS.

The MQ IMS adapter is the older method to connect MQ to IMS. This method requires updating the IMS transaction to include MQ calls.

By using the MQ IMS bridge, you can reengineer applications that were controlled by 3270-connected terminals to be controlled by MQ messages, without having to rewrite, recompile, or relink them. The bridge is an IMS Open Transaction Manager Access (OTMA) client.

IBM Training

IBM MQ IMS bridge



- IMS application gets its input by using a GET UNIQUE (GU) call to the IOPCB class and sends its output by using an INSERT (ISRT) call to the IOPCB
- MQ applications use the IMS header in the message to ensure that the applications can run as they did when driven by nonprogrammable terminals
- Queue manager can connect to one or more IMS systems
- More than one queue manager can connect to one IMS system if they all belong to the same XCF group and are in the same sysplex

Figure 9-3. IBM MQ IMS bridge

The MQ IMS OTMA bridge is the most practical way of integrating MQ applications with IMS transactions.

When the MQ IMS OTMA bridge is used, the IMS GU gets the data. OTMA does not require any additional work in the older IMS transaction code.

In z/OS, IMS automatically consumes any messages that are defined to use the MQ IMS OTMA bridge. The bridge requires an IMS trigger monitor transaction to start the required IMS transaction.

When you use the MQ IMS OTMA bridge to send messages to an IMS application, you need to construct your messages in a special format. You must also put your messages on MQ queues that are defined with a storage class that specifies the z/OS cross-system coupling facility (XCF) group and member name of the target IMS system.

IBM Training

IBM MQ and CICS

- Use MQ with CICS by configuring the CICS adapter and CICS bridge
- CICS adapter
  - Connect a queue manager to CICS
  - CICS applications can use the MQI
- IBM MQ CICS bridge (optional)
  - Applications can run a CICS program or transaction that does not use the MQI
  - Use existing applications with MQ, without rewriting them
  - If requested, the bridge can check the user ID and password extracted from the MQ request message before running the CICS program named in the request message

Figure 9-4. IBM MQ and CICS

To use MQ with CICS, you must configure the IBM MQ CICS adapter and the IBM MQ CICS bridge components.

The CICS to MQ adapter is supplied with CICS. It communicates with MQ as an external resource manager by using the CICS Resource Manager Interface.

The CICS to MQ adapter provides a set of control functions for use by system programmers and administrators to manage the adapter. It also provides MQ MQI support for CICS applications.

The optional IBM MQ CICS bridge enables applications to run a CICS program or transaction that does not use the MQI. So, you can use your existing applications with MQ, without the need to rewrite them.

If requested, the bridge can check the user ID and password from the MQ message before it runs the CICS program that is named in the message. The queue manager uses an external security manager such as RACF to do authentication. User IDs in the request message must be defined to the external security manager.

IBM Training

IBM MQ CICS bridge

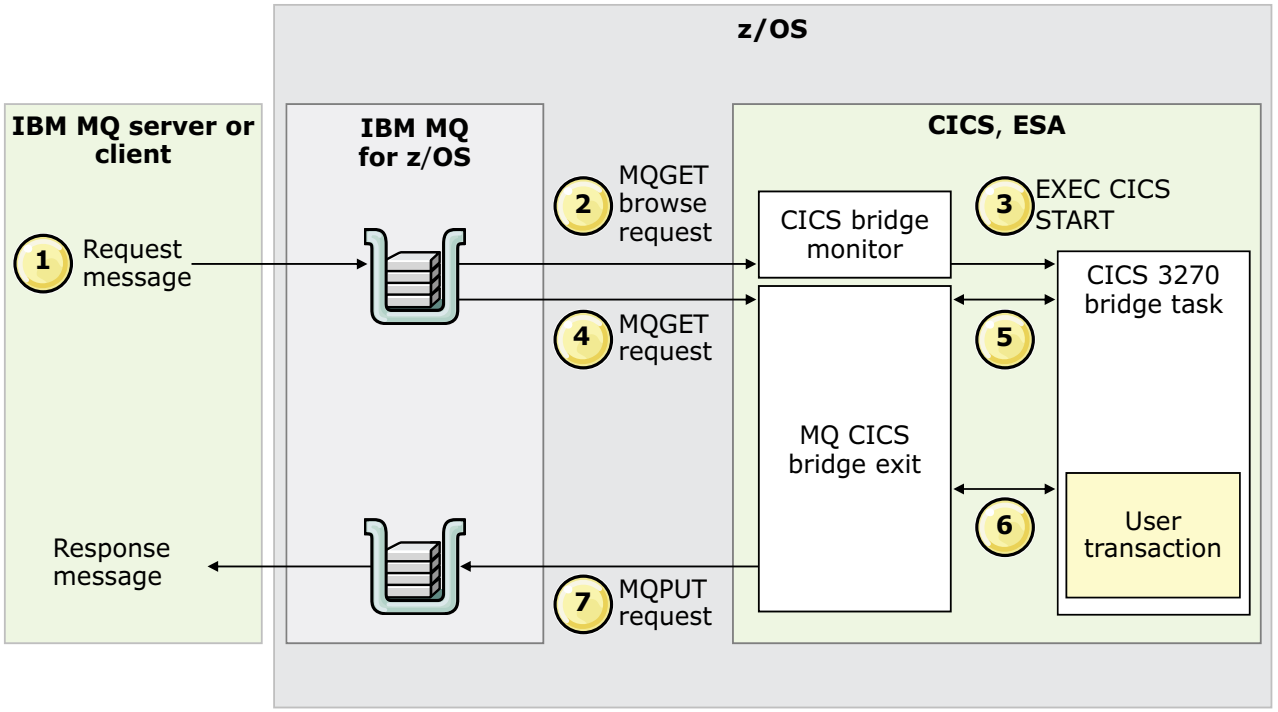


Figure 9-5. IBM MQ CICS bridge

The MQ CICS bridge allows an application to run a 3270-based transaction without knowledge of the 3270 data stream. Data that is necessary to run the transaction is put on a queue by a program that is not a CICS program. From the CICS transaction perspective, it runs as though it has a real 3270 terminal. In reality, MQ messages are used for communication.

Unlike traditional emulators, the bridge does *not* replace VTAM flows with MQ messages. The program is dealing with the transaction rather than with an emulator.

IBM Training

JMS support in IBM MQ

- IBM MQ for JMS and JMS are both message-oriented middleware
- IBM MQ Classes for JMS is a JMS provider that implements the JMS interfaces for IBM MQ
- MQ Explorer supports JMS object administration
  - Queues
  - Connection factories
  - Destinations

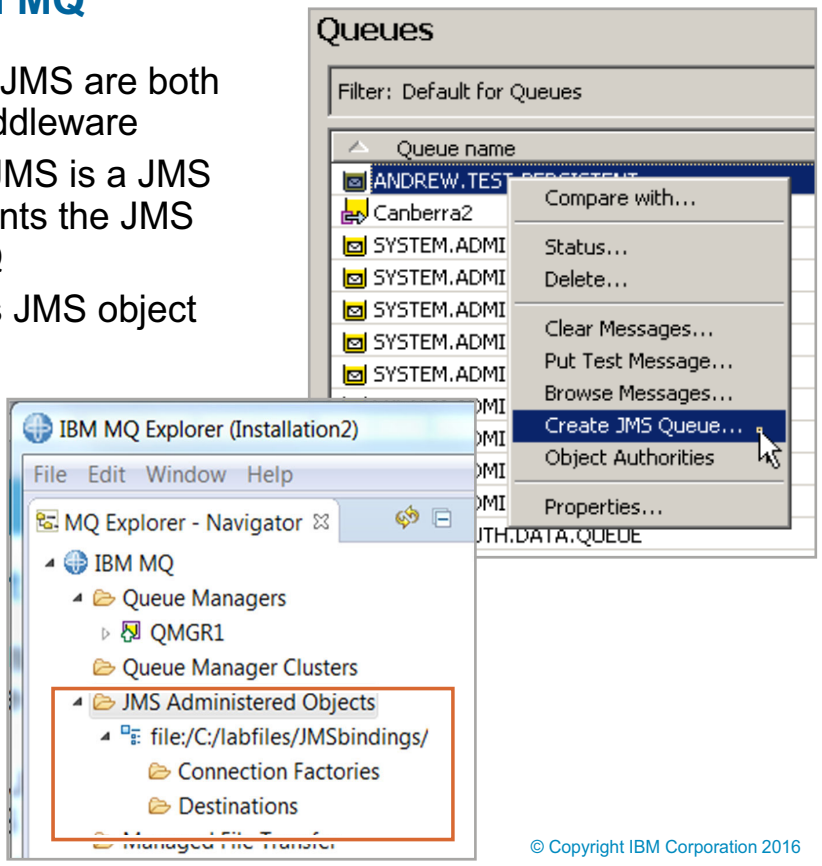


Figure 9-6. JMS support in IBM MQ

IBM MQ classes for JMS is the JMS provider that is supplied with MQ.

The JMS specification expects Connection Factory and Destination objects to be administered objects. An administrator creates and maintains administered objects in a central repository, and a JMS application retrieves these objects by using the Java Naming Directory Interface (JNDI).

IBM MQ classes for JMS supports the use of administered objects. An administrator can use either the IBM MQ JMS administration tool or MQ Explorer to create and maintain administered objects.

IBM MQ classes for JMS

- Encapsulate MQI
- Easy to implement for anyone familiar with MQI in other languages
- Can take advantage of the full features of MQ
- Similar object model to other object-oriented language interfaces such as C++ and .NET

Figure 9-7. IBM MQ classes for JMS

Using IBM MQ classes for JMS has a number of advantages:

- You can reuse any existing JMS skills in your organization.
- Connection factories and destinations can be stored as JMS administered objects in a central repository rather than being hardcoded into an application. Applications are more independent from the JMS provider and the underlying MQ configuration.
- IBM MQ classes for JMS provide access to both the point-to-point and publish/subscribe messaging features of IBM MQ.

The MQ classes JMS have a close correlation to the underlying MQI. If you are familiar with programming in the MQI, you should have little trouble adapting that knowledge to the classes for JMS.

IBM Training

IBM JMS extensions

- Provide a greater level of consistency across IBM JMS providers
- Can be used with any messaging provider
- Use these extensions to:
  - Create connection factories and destinations dynamically at run time instead of retrieving them from JNDI namespace
  - Set the properties of MQ classes for JMS objects
  - Obtain detailed information about a problem
  - Control tracing
  - Obtain version information about IBM MQ Classes for JMS

Figure 9-8. IBM JMS extensions

IBM MQ classes for JMS contain a set of extensions to the JMS API called the IBM JMS extensions. An application can use these extensions to create connection factories and destinations dynamically at run time, and to set the properties of IBM MQ classes for JMS objects.

The JMS extensions provide the following functions:

- A factory-based mechanism for creating connection factories and destinations dynamically at run time, instead of retrieving them as administered objects from a Java Naming and Directory Interface (JNDI) namespace.
- A set of methods for setting the properties of IBM MQ classes for JMS objects.
- A set of exception classes with methods for obtaining detailed information about a problem.
- A set of methods for controlling tracing.
- A set of methods for obtaining version information about IBM MQ classes for JMS.

The JMS extensions can be used with any messaging provider.

IBM Training

IBM MQ resource adapter

- Allows applications that are running in an application server to access MQ resources
  - **Outbound communication**: Application starts a connection to a queue manager, sends JMS messages to JMS destinations, and receives JMS messages from JMS destinations in a synchronous manner
  - **Inbound communication**: JMS message that arrives at a JMS destination is delivered to a message driven bean, which processes the message asynchronously
- Installation
  - Can install on any application server that is certified as compliant with the Java Platform, Enterprise Edition 7 specification and supports JMS 2.0
  - Required to connect to MQ from WebSphere Application Server Liberty
  - Preinstalled within WebSphere Application Server traditional Version 9.0

Figure 9-9. IBM MQ resource adapter

The IBM MQ resource adapter allows applications that are running in an application server to access MQ resources. It supports inbound and outbound communication.

The MQ resource adapter implements the Java Platform, Enterprise Edition Connector Architecture interfaces and contains the IBM MQ classes for JMS. It allows JMS applications and message driven beans that run in an application server to access the resources of an MQ queue manager. The resource adapter supports both the point-to-point domain and the publish/subscribe domain.

The IBM MQ resource adapter supports inbound and outbound communication.

You can install the resource adapter on any application server that is certified as compliant with the Java Platform, Enterprise Edition 7 specification.

The MQ resource adapter can be deployed only into an application server that supports JMS 2.0.

IBM Training

Using IBM MQ with WebSphere Application Server

- MQ can be used with, or as an alternative to, the default messaging provider that is included with WebSphere Application Server
- Ways to integrate with WebSphere Application Server:
  - MQ as a JMS provider
  - MQ messaging provider
  - MQ resource adapter

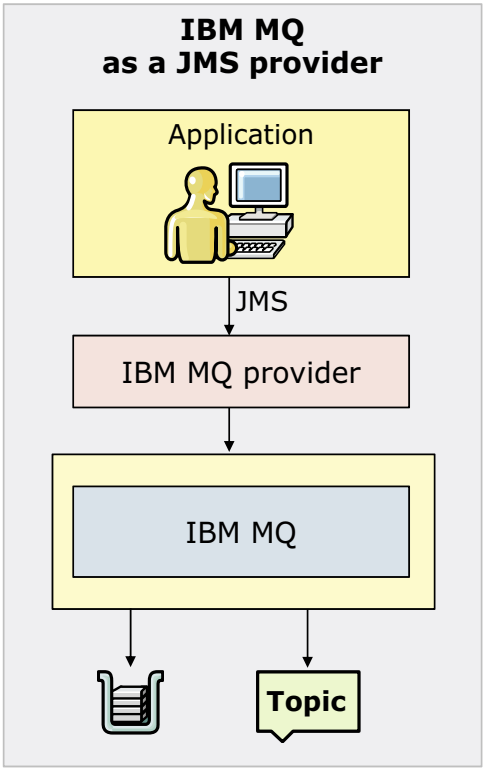


Figure 9-10. Using IBM MQ with WebSphere Application Server

IBM MQ can be used with, or as an alternative to, the default messaging provider that is included with WebSphere Application Server.

The MQ messaging provider is installed as part of WebSphere Application Server Version 8.5. This provider includes a version of the MQ resource adapter, and the MQ Extended Transactional Client functions, which allows the queue manager to participate in XA transactions that are managed by the application server.

When you use MQ as a JMS messaging provider, MQ supports JMS domain-independent interfaces. Applications can use the same interface when they use point-to-point or publish/subscribe messaging styles within the same transaction.

Introducing IBM Integration Bus

- Single product for IBM MQ, Windows .NET, Java, and fully heterogeneous integration scenarios
- Provides connectivity across enterprise systems, applications, and data
  - Avoids rewrites in response to new integration requirements
  - Simplifies maintenance by reducing expensive coupling
  - Provides flexibility, which adds anonymity between data producers and consumers
  - Adds insight into applications and the business value that they bring

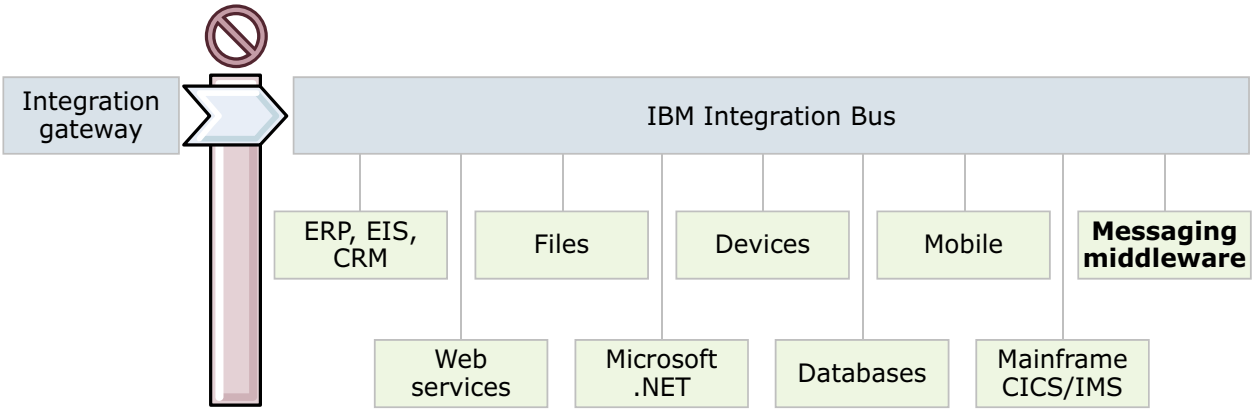


Figure 9-11. Introducing IBM Integration Bus

IBM Integration Bus delivers a comprehensive integration solution across the enterprise.

Integration Bus connects a wide range of applications, services, and systems across heterogeneous IT environments. It provides the visibility and control capabilities that are needed to support critical business activities such as monitoring, auditing, process management, and analytics.

Integration Bus can integrate with MQ and help to do the following tasks:

- Rapidly enable business insight to be applied to in-flight data
- Accelerate creation of integration services for business process management
- Increase operational awareness and control over workload
- Gain visibility and insight of integration in application environments

IBM Training

IBM MQ connectivity options for IBM Integration Bus

- Integration Bus message flow applications can access existing MQ networks to get and put messages to application queues
  - Can connect to local or remote queue managers
  - Can receive messages from multiple queues on multiple queue managers
- Policy and connection properties define links to queue managers
- Integration Bus license includes entitlement to MQ

Figure 9-12. IBM MQ connectivity options for IBM Integration Bus

You can use Integration Bus message applications to access existing MQ networks. By using MQ message processing nodes, Integration Bus connects to MQ as a client and can put and get messages to MQ queues.

For flexibility, the Integration Bus administrator can define connections to MQ by using connection policies. Connect policies allow the administrator to change MQ connectivity without changing the Integration Bus runtime applications.

An Integration Bus license includes entitlement to MQ.

IBM Training

IBM Integration Bus features that require IBM MQ

- z/OS implementations
- Recording and replaying transaction events
- Global transaction management
- Event-driven message processing nodes that are used for aggregation and timeout flows, message collections, and message sequences
- IBM MQ File Transfer Edition message processing
- IBM Sterling Connect:Direct message processing
- IBM Business Process Manager Advanced nodes that use MQ bindings
- SAP with transactional message processing
- Integration nodes with HTTP listeners
- HTTP proxy servlet
- High availability configurations

Figure 9-13. IBM Integration Bus features that require IBM MQ

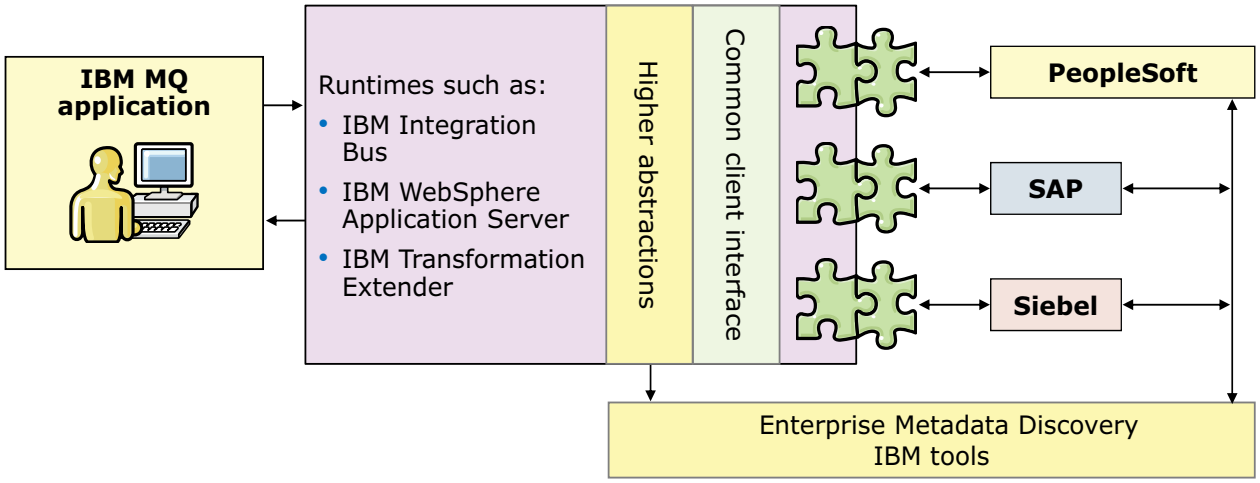
Some Integration Bus features require connectivity to an MQ queue manager.

On z/OS, MQ is required for installation, and only local connections to queue managers are supported. You must specify a queue manager for the Integration Bus runtime engine (integration node), but you can also connect to other local queue managers on MQ message flow nodes by using server bindings for the connection.

Many of IBM Integration Bus features that are listed in the figure require access to system queues on an MQ queue manager for the storage and retrieval of state information.

IBM Training

IBM WebSphere Adapters



- Configurable software components provide connectivity to enterprise applications
- Tools provide metadata discovery by using JCA standards
- Transactional two-way interaction with application without coding an interface
- Application adapters such as PeopleSoft, SAP, and Siebel
- Technology adapters such as flat file, email, JDBC, IBM i, and Lotus Domino
- Specific adapter availability varies by runtime selection

Expanding the scope of IBM MQ

Figure 9-14. IBM WebSphere Adapters

You can connect IBM MQ applications to enterprise information systems such as SAP, PeopleSoft, and Siebel, by using an IBM application and IBM WebSphere Adapters. These IBM applications include IBM Integration Bus, IBM Transformation Extender, and IBM WebSphere Application Server.

IBM WebSphere Adapters provide JCA-compliant mechanisms to extend the enterprise. WebSphere Adapters help you minimize the need for integration coding and create standard interface points. WebSphere Adapters help extend service-oriented architecture applications beyond organizational walls to customers, partners, and suppliers. They can reduce maintenance and development costs while optimizing and renewing the value of your enterprise assets.

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 PureApp 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

Figure 9-15. IBM MQ in the Cloud

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

IBM Training

Cloud technology checkpoint

- Hypervisors
  - Virtualize the operating system and hardware where they run
  - Divide the resources of a server across a number of virtual machines
  - Manage state of virtual machines on the physical server
- Examples of hypervisors are:
  - PowerVM for AIX
  - VMware ESX hypervisor
  - IBM PureApplication System also contains a hypervisor

Figure 9-16. Cloud technology checkpoint

Virtualization is the creation of flexible substitutes for actual resources. Virtualization is commonly applied to physical hardware resources by combining multiple physical resources into shared pools from which users receive virtual resources. With virtualization, you can make one physical resource look like multiple virtual resources.

System virtualization creates many virtual systems within a single physical system. Virtual systems are independent operating environments that use virtual resources.

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

IBM Training

IBM MQ Hypervisor editions

- Contain the operating system and an IBM MQ installation
  - IBM MQ Hypervisor Edition for Red Hat Enterprise Linux Server
  - IBM MQ Hypervisor Edition for AIX Editions
- 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

Figure 9-17. IBM MQ Hypervisor editions

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

IBM MQ Advanced for PureApplication

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

Figure 9-18. 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 pre-configured 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 MQ Advanced PureApplication pattern can be managed by the PureApplication console 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
  - MQ Light
  - REST
  - Kafka
- Supports batch and real-time analytics

Figure 9-19. 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 has 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 Message Connect on IBM Bluemix

- Connects enterprise MQ message network to the cloud
  - Scalable, distributed, high throughput message bus
  - Ingests data into analytics services to optimize business decisions
  - Builds on MQ's dominance in enterprise messaging
- Combination of Message Hub and Message Connect provides 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

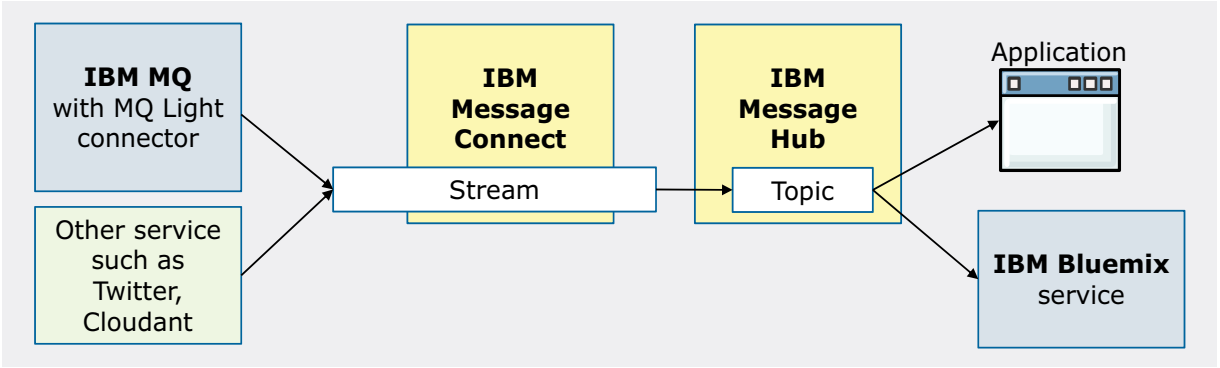


Figure 9-20. IBM Message Connect on IBM Bluemix

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 pre-built to connect to data sources and flow-specific data points on to 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 MQ queue manager inside a Docker container managed by Linux kernel
  - Provides process isolation, resource, isolation, and dependency isolation
- Chef cookbook for IBM MQ
  - Source available on GitHub
  - Installs MQ server and client
  - Creates and starts a queue manager
  - Calls MQSC to define a queue
- IBM MQ on Openstack
  - Create custom virtual images

Figure 9-21. 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 you allocate to a container.
- All the software on which MQ depends on 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

Unit summary

- Identify the IBM MQ z/OS links and bridges
- Describe how IBM MQ can be used as part of the communications infrastructure to act as a JMS provider
- Describe how IBM MQ interfaces with IBM Integration Bus
- Distinguish ways that IBM MQ is used by WebSphere Application Server
- Describe the options for deploying IBM MQ in a Cloud environment

Figure 9-22. Unit summary

Review questions

1. What are some advantages of using the IBM MQ IMS bridge over the IBM MQ IMS adapter?
   A. The IMS transaction starts in wait-for-input (WFI) mode
   B. The application needs to code an OTMA client only
   C. The IMS GU obtains the message payload
   D. The OTMA bridge is configured between IBM MQ and IMS, and no changes need to be made to the IMS transaction

2. The IBM MQ Hypervisor Editions contains which of the following components?
   A. The operating system and an IBM MQ installation
   B. A preconfigured MQ cluster and the operating system
   C. The operating system and an installation of IBM Integration Bus

Figure 9-23. Review questions

Write your answers here:

1.

2.

Review answers

1. What are some advantages of using the IBM MQ IMS bridge over the IBM MQ IMS adapter?
   A. The IMS transaction starts in wait-for-input (WFI) mode
   B. The application needs to code an OTMA client only.
   C. The IMS GU obtains the message payload
   D. The OTMA bridge is configured between IBM MQ and IMS, and no changes need to be made to the IMS transaction

   The answer is C and D.

2. The IBM MQ Hypervisor Editions contains which of the following components?
   A. The operating system and an IBM MQ installation
   B. A pre-configured IBM MQ cluster and the operating system
   C. The operating system and an installation of IBM Integration Bus

   The answer is A.

Figure 9-24. Review answers

Unit 10. Course summary

Estimated time

00:15

Overview

This unit summarizes the course and provides information for future study.

IBM Training

Unit objectives

- Explain how the course met its learning objectives
- Access the IBM Training website
- Identify other IBM Training courses that are related to this topic
- Locate appropriate resources for further study

Figure 10-1. Unit objectives

IBM Training

Course objectives

- Summarize current business drivers and the need for flexibility
- Describe enterprise messaging and the capabilities it must provide
- Identify the main ways that IBM MQ can impact application design
- Describe the basic components of IBM MQ
- Differentiate between point-to-point and IBM MQ cluster connectivity
- Summarize queue manager and queue manager components administrative tasks
- Contrast the architectural role of IBM MQ clusters and multiple instance queue managers

Figure 10-2. Course objectives

IBM Training

Course objectives

- Describe the security provisions of IBM MQ and IBM MQ Advanced Message Security
- Describe how IBM MQ is used as part of the communications infrastructure to:
  - Connect application environments, such as the World Wide Web, enterprise transaction systems, and database systems
  - Manage the distribution of publisher information to appropriate subscribers
  - Provide file transfer management with IBM MQ Managed File Transfer
  - Serve as a JMS provider
  - Interface with WebSphere Application Server
  - Store in-flight messages for IBM Integration Bus
  - Interact with z/OS applications
  - Facilitate connectivity to mobile environments with IBM MQ Telemetry

Figure 10-3. Course objectives

IBM Training

Course objectives

- Describe the options for deployment to the Cloud

Figure 10-4. Course objectives

IBM Training

To learn more on the subject

- IBM Training website:
  www.ibm.com/training
- IBM Knowledge Center for IBM MQ V9.0:
  www.ibm.com/support/knowledgecenter/SSFKSJ_9.0.0/com.ibm.mq.helphome.v90.doc/WelcomePagev9r0.htm

Figure 10-5. To learn more on the subject

IBM Training

Enhance your learning with IBM resources

Keep your IBM Cloud skills up-to-date

- IBM offers resources for:
  - Product information
  - Training and certification
  - Documentation
  - Support
  - Technical information

- To learn more, see the IBM Cloud Education Resource Guide:
  - www.ibm.biz/CloudEduResources

Figure 10-6. Enhance your learning with IBM resources

IBM Training

Unit summary

- Explain how the course met its learning objectives
- Access the IBM Training website
- Identify other IBM Training courses that are related to this topic
- Locate appropriate resources for further study

Figure 10-7. Unit summary

IBM Training

Course completion

You have completed this course:

Technical Introduction to IBM MQ

Any questions?

Figure 10-8. Course completion

Appendix A. List of abbreviations

ACL	Access Control List
AMQP	Advanced Message Queuing Protocol
API	Application programming interface
ASCII	American Standard Code for Information Interchange
ATM	Automatic Teller Machine
CCDT	Client channel definition table
CDR	Continuous Delivery Release
CF	coupling facility
COBOL	Common Business Oriented Language
CPU	central processing unit
DMZ	demilitarized zone: A configuration that includes multiple firewalls to add layers of protection between a corporate intranet and a public network, such as the Internet.
EAR	enterprise archive
EBCDIC	Extended Binary Coded Decimal Interchange Code
FDC	first failure data capture
FTP	File Transfer Protocol
FTPS	File Transfer Protocol over Secure Socket Layer
GPFS	General Parallel File System
HA	high availability
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IaaS	Infrastructure as a Service
IBM	International Business Machines Corporation
IFS	Integrated File System
IOPCB	IMS I/O Program Communication Block
IoT	Internet of Things
IP	Internet Protocol
ISPF	Interactive System Productivity Facility
IT	information technology
JAAS	Java Authentication and Authorization Service
Java EE	Java Platform, Enterprise Edition

JCA	Java EE Connector Architecture
JDBC	Java Database Connectivity
JMS	Java Message Service
JNDI	Java Naming and Directory Interface
JRE	Java runtime environment
KVM	Kernel-based Virtual Machine
LTSR	Long-Term Support Release
M2M	Machine-to-machine
MCA	Message channel agent
MOM	Message-oriented middleware
MQ	Message queue
MQAI	MQ Administration Interface
MQI	IBM MQ Message queue
MQIPT	IBM MQ Internet Pass-Thru
MQMD	MQ message descriptor
MQMFT	MQ Managed File Transfer
MQSC	IBM MQ script commands
MQTT	MQ Telemetry Transport
OAM	Object authority manager
OS	operating system
OTMA	Open Transaction Manager Access
PaaS	Platform as a Service
PC	personal computer
PCF	Programmable Command Format
QA	Quality Assurance
QoS	Quality of service
RACF	Resource Access Control Facility
REST	Representational State Transfer
RFID	radio frequency identification
RSMB	Really Small Message Broker
SAF	System Authorization Facility
SFTP	Secure File Transfer Protocol
SOA	service-oriented architecture
SOCKS	Sockets Secure

SSD	Solid state drive
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TM	transaction manager
URL	Uniform Resource Locator
VTAM	Virtual Telecommunications Access Method
WFI	wait-for-input
XA	extended architecture
XCF	Cross-system coupling facility
XML	Extensible Markup Language

IBM Training