Introduction to IBM MQ

Roger Lacroix
roger.lacroix@capitalware.com
http://www.capitalware.com
Why use IBM MQ

Things to consider:

- Importance of parallel operations
- Variety of info flow patterns
- Dependence upon serialization
- Message traffic volumes
- Relationship of programs
- Programming skills level
- Knowledge of networking
- Mixtures of application types
- Mixed old/new applications
Queuing Technology

External Network
Reasons to Choose IBM MQ

- Take away the communications nightmares
- Component-built applications approach
- Allow control of load balancing
- Allow protocol independence
- Provide consistent programming interface across platforms
- More supported platforms than any other product
Business Perspective

- Time independent (asynchronous) processing
- Connectionless communications
- Assured message delivery
- Once and once only delivery
- Syncpoint control
- Resource manager
- Integrated with operating system
- Interfaces to other system managers
- Triggering, Message grouping, Clustering
Application Perspective

- Single, multi-platform Application Programming Interface (API)
- Faster application development
- Portable code
MQ Environment Overview

API (MQI)

Program A
Queue Manager
Handles
Queues

Program B

Program C
Queue Manager
Handles
Queues

Program D

Program E

Q Q Q Q

Q Q Q Q
IBM MQ Clients

- An IBM MQ client is a component that can be installed on a separate machine from the Base product and Server.
- IBM MQ applications can run on client.
- Client uses a server queue manager via a network protocol (i.e. TCP/IP).
Why use IBM MQ Clients?

- Supported on roughly 45 platforms
- Reduces client hardware requirements
- IBM MQ Client license is free
Connecting: Bindings vs Client Mode

Server Model
- Application
- MQ Server Library
- MQ Server
- Inter process Communications
- Local or bindings mode

Client Model
- Application
- MQ Client Library
- MQ Server
- Network Communications
- Client mode
Terminology

Message:

A message is a collection of data sent by one program and intended for another program.
Examples of Messages

- Units of information transfer (one-way)
- A request for service or information
- A reply to a service or information request
- A report of status
- An announcement or broadcast
MQ Message Overview

Message = Header + User Properties + User Data

- **Header**
  - A Series of Message Attributes
  - Understood and augmented by the Queue Manager
  - Message Id
  - Correlation Id
  - Routing information
  - Reply routing information
  - Message priority
  - Message codepage/encoding
  - Message format
  - etc.

- **User Properties**
  - Any sequence of bytes
  - Private to the sending and receiving programs
  - Not meaningful to the Queue Manager

- **User Data**
  - User Properties require WMQ V7
    - Emulated for JMS in older versions of WMQ
  - Arbitrary properties
    - For example, this is a "green" message
Terminology

MQ Objects:

- Queue managers
- Queues
- Channels
- Processes
- Namelists
- Distribution Lists
- Topics
- Storage Classes
Terminology

Queue Manager:

A queue manager is the IBM MQ component that provides the messaging and queuing services to application programs through Message Queue Interface (MQI) program calls.
Queue Manager Characteristics

May define Multiple Queue Managers
- Development
- Testing
- Acceptance
- Production

Name
- Up to 48 characters in length, case sensitive
- Z/OS only 4 characters (Subsystem ID)
Terminology

Queue:

A queue is an IBM MQ object that can store messages. It has attributes that determine what processing occurs when an application accesses it through the MQI calls.
Queue Definition Types

- Local
- Remote
- Alias
- Model

- Dynamic:
  - Permanent
  - Temporary

- System:
  - Transmission
  - Event
  - Dead letter
  - Initiation
  - Command server
  - Cluster
  - Pub/Sub Broker Queues

Capitalware's MQ Technical Conference v2.0.1.4
Terminology

Channel:

A channel is a communication link providing a path on the same or different platforms. The message channel is used for the transmission of messages from one queue manager to another, and shields the application programs from the complexities of the underlying networking protocols.
Channel Characteristics

- Sender - Receiver - push-type model
- Requester - Server - pull-type model
- Requester - Sender - call-back model
- Server - Receiver - push-type model
- Cluster Server – Cluster Receiver - push-type model

Note: Channel pairs must match, names of channels in pairs must be identical
Terminology

Process:

A process is an IBM MQ object that defines an application to the IBM MQ Queue Manager.
Process Characteristics

- Process definition used to identify applications to be started by a trigger monitor
- The process definition includes application ID and type, plus some application specific data
Terminology

Namelist:

A Namelist is an IBM MQ object that contains a list of other IBM MQ objects.
Namelist Characteristics

- Example of use is for trigger monitors where a Namelist could contain a list of queues to monitor.

- Can be maintained independently of applications that use it.
Terminology

**Topic:**

A topic is an IBM MQ object that contains a theme that applications publish messages to.
Topic Characteristics

- A topic string is a case sensitive
- '/' The topic level separator – provides structure to topic trees
- '#' The wildcard character
- '+' The single-level wildcard character

The Topic can be defined in a number of ways:
- Predefined by the MQSC command
- Predefined by the PCF interface (as used by the IBM MQ Explorer)
- Subscribing or Publishing to the Topic object
Terminology

Distribution List:

A Distribution List is used by an IBM MQ application to access a group IBM MQ queues. It is generated by the application.
Distribution List Characteristics

- Allows one MQPUT to send a message to many destinations.

- If the same transmission queue is to be used for multiple destinations, only one copy of the message is placed on the queue.
Terminology

Storage Class:

A Storage Class is an IBM MQ object that is used to map one or more queues to a OS/390 page set.
Queue/STGCLASS Relationship

- QUEUE(A)
  STGCLASS(ALPHA)

- QUEUE(B)
  STGCLASS(ALPHA)

- QUEUE(C)
  STGCLASS(BETA)

Storage Class
“ALPHA”

- Page Set 1
  Message 1
  Message 2
  etc...

Storage Class
“BETA”

- Page Set 2
  Message 3
  Message 4
  etc...
Queue Manager Clusters

- This is a logical grouping of Queue Managers (QMGRs). The QMGRs can be physically remote.

- A QMGR can belong to more than one cluster. (Overlapping clusters)

- QMGRs can advertise Qs to the clusters. Channel definitions, XMITQ definitions, and QR definitions are not required.

- More than one QMGR can advertise the same Q name. (work load balancing / failover)
Message Queue Interface (Procedural)

- 13 verbs (original) + 12 introduced in WMQ v7.0
- 7 are used most commonly
- Important to understand use of call parameters
Common MQI Calls

- **MQCONN** Connect to Queue Manager
- **MQOPEN** Open a queue
- **MQPUT** Put message to queue
- **MQGET** Get message from queue
- **MQCLOSE** Close a queue
- **MQDISC** Disconnect from Queue Manager
- **MQPUT1** Put one message on a queue
Specialized MQI Calls

- MQBEGIN: Signals start of Unit of Work
- MQCMIT: Commits Unit of Work
- MQBACK: Rollback Unit of Work
- MQINQ: Inquire on MQ object
- MQSET: Set queue attributes
- MQCONNX: Connect with special options
New MQI Calls in WMQ v7.0

- MQCB - Defines a callback function
- MQCTL - Start/stop message delivery
- MQSUB - Registers a subscription
- MQSUBRQ - Request services from a subscription
- MQSTAT - Obtain information about previous Async puts
New MQI Calls in WMQ v7.0

- MQSETMP - Set a message property
- MQINQMP - Inquire on a message property
- MQDLTMP - Delete a message property
- MQCRTMH - Create a message handle
- MQDLMTH - Delete a message handle
- MQMHBUF - Converts buffer into message handle
- MQBUFMH - Converts a message handle into a buffer
# Summary of MQ API Verbs

<table>
<thead>
<tr>
<th>Connection</th>
<th>Resource Use</th>
<th>Messages</th>
<th>Object attributes</th>
<th>Transactions</th>
<th>Message Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>MQCONN</td>
<td>MQOPEN</td>
<td>MQPUT</td>
<td>MQINQ</td>
<td>MQBEGIN</td>
<td>MQCRTMH</td>
</tr>
<tr>
<td>MQCONNX</td>
<td>MQSUB</td>
<td>MQPUT1</td>
<td>MQSET</td>
<td>MQCMIT</td>
<td>MQCLTMH</td>
</tr>
<tr>
<td>MQCTL</td>
<td>MQSUBRQ</td>
<td>MQGET</td>
<td></td>
<td>MQBACK</td>
<td>MQSETMP</td>
</tr>
<tr>
<td>MQDISC</td>
<td>MQCLOSE</td>
<td>MQCB</td>
<td></td>
<td></td>
<td>MQINQMP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MQDLTMP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MQMHBUF/MQBFUMH</td>
</tr>
</tbody>
</table>
Application/MQ Relationship

Non-trusted

- Application
- MQI Stub
- IPCC

MQCONN MQDISC

IPCC

Execution Controller

MQOPEN MQCLOSE MQPUT MQGET etc.

IPCC

LQM Agent

Queue Manager
Shared Resources
Application/MQ Relationship

Trusted

MQCONNX
MQDISC

Execution Controller

Queue Manager
Shared Resources

Application

MQI Stub

LQM Agent

MQOPEN
MQCLOSE
MQPUT
MQGET
e tc.

Capitalware's MQ Technical Conference v2.0.1.4
PCF Messages

- Specialized Message Structure, similar to that used in MQ Event messages.
- Header and multiple parameters
- Has similar functions to RUNMQSC commands
- Permits Remote Management
The Big Picture

QM 1
Program A
Put Q2
MQI
Put Q1
Program B
Get Q1

QM 2
Program C
Get Q2
MQI

Message and Queuing
QM 2 XmitQ
Q1

TCP/IP, APPC etc

Channel
Questions & Answers
Why use IBM MQ

Things to consider:

- Importance of parallel operations
- Variety of info flow patterns
- Dependence upon serialization
- Message traffic volumes
- Relationship of programs
- Programming skills level
- Knowledge of networking
- Mixtures of application types
- Mixed old/new applications
Queuing Technology
Reasons to Choose IBM MQ

- Take away the communications nightmares
- Component-built applications approach
- Allow control of load balancing
- Allow protocol independence
- Provide consistent programming interface across platforms
- More supported platforms than any other product
**Business Perspective**

- Time independent (asynchronous) processing
- Connectionless communications
- Assured message delivery
- Once and once only delivery
- Syncpoint control
- Resource manager
- Integrated with operating system
- Interfaces to other system managers
- Triggering, Message grouping, Clustering
Application Perspective

- Single, multi-platform Application Programming Interface (API)
- Faster application development
- Portable code
IBM MQ Clients

- An IBM MQ client is a component that can be installed on a separate machine from the Base product and Server.
- IBM MQ applications can run on client.
- Client uses a server queue manager via a network protocol (i.e. TCP/IP).
Why use IBM MQ Clients?

- Supported on roughly 45 platforms
- Reduces client hardware requirements
- IBM MQ Client license is free
Connecting: Bindings vs Client Mode

Server Model
- Application
- MQ Server Library
- Inter process Communications
- MQ Server
- Local or bindings mode

Client Model
- Application
- MQ Client Library
- Network Communications
- MQ Server
- Client mode
Terminology

Message:
A message is a collection of data sent by one program and intended for another program.
Examples of Messages

- Units of information transfer (one-way)
- A request for service or information
- A reply to a service or information request
- A report of status
- An announcement or broadcast
MQ Message Overview

Message = Header + User Properties + User Data

- Header
  - A Series of Message Attributes
  - Understood and augmented by the Queue Manager
  - Message Id
  - Correlation Id
  - Routing Information
  - Reply-to routing Information
  - Message priority
  - Message codepage/encoding
  - Message format
  - etc.

- User Properties
  - Any sequence of bytes
  - Private to the sending and receiving programs
  - Not meaningful to the Queue Manager

- User Data
  - User Properties require WMQ V7
  - Emulated for JMS in older versions of WMQ
  - Arbitrary properties
  - For example, this is a “green” message
**Terminology**

**MQ Objects:**
- Queue managers
- Queues
- Channels
- Processes
- Namelists
- Distribution Lists
- Topics
- Storage Classes
Terminology

Queue Manager:

A queue manager is the IBM MQ component that provides the messaging and queuing services to application programs through Message Queue Interface (MQI) program calls.
Queue Manager Characteristics

- May define Multiple Queue Managers
  - Development
  - Testing
  - Acceptance
  - Production

- Name
  - Up to 48 characters in length, case sensitive
  - Z/OS only 4 characters (Subsystem ID)
**Terminology**

**Queue:**

A queue is an IBM MQ object that can store messages. It has attributes that determine what processing occurs when an application accesses it through the MQI calls.
Queue Definition Types

- Local
- Remote
- Alias
- Model
- Dynamic:
  - Permanent
  - Temporary
- Cluster

System:
- Transmission
- Event
- Dead letter
- Initiation
- Command server
- Cluster
- Pub/Sub Broker Queues
**Terminology**

**Channel:**

A channel is a communication link providing a path on the same or different platforms. The message channel is used for the transmission of messages from one queue manager to another, and shields the application programs from the complexities of the underlying networking protocols.
## Channel Characteristics

- Sender - Receiver - push-type model
- Requester - Server - pull-type model
- Requester - Sender - call-back model
- Server - Receiver - push-type model
- Cluster Server – Cluster Receiver - push-type model

Note: Channel pairs must match, names of channels in pairs must be identical
**Terminology**

**Process:**

A process is an IBM MQ object that defines an application to the IBM MQ Queue Manager.
Process Characteristics

- Process definition used to identify applications to be started by a trigger monitor
- The process definition includes application ID and type, plus some application specific data
Terminology

Namelist:

A Namelist is an IBM MQ object that contains a list of other IBM MQ objects.
Namelist Characteristics

- Example of use is for trigger monitors where a Namelist could contain a list of queues to monitor
- Can be maintained independently of applications that use it
Terminology

Topic:

A topic is an IBM MQ object that contains a theme that applications publish messages to.
Topic Characteristics

- A topic string is a case sensitive
- '/' The topic level separator – provides structure to topic trees
- '#' The wildcard character
- '+' The single-level wildcard character

The Topic can be defined in a number of ways:
- Predefined by the MQSC command
- Predefined by the PCF interface (as used by the IBM MQ Explorer)
- Subscribing or Publishing to the Topic object
Terminology

Distribution List:

A Distribution List is used by an IBM MQ application to access a group IBM MQ queues. It is generated by the application.
Distribution List Characteristics

- Allows one MQPUT to send a message to many destinations.
- If the same transmission queue is to be used for multiple destinations, only one copy of the message is placed on the queue.
Terminology

**Storage Class:**

A Storage Class is an IBM MQ object that is used to map one or more queues to a OS/390 page set.
Queue/STGCLASS Relationship

- QUEUE(A) STGCLASS(ALPHA)
- QUEUE(B) STGCLASS(ALPHA)
- QUEUE(C) STGCLASS(BETA)

Storage Class “ALPHA”
Page Set 1
Message 1
Message 2
etc...

Storage Class “BETA”
Page Set 2
Message 3
Message 4
etc...
Queue Manager Clusters

- This is a logical grouping of Queue Managers (QMGRs). The QMGRs can be physically remote.
- A QMGR can belong to more than one cluster. (Overlapping clusters)
- QMGRs can advertise Qs to the clusters. Channel definitions, XMITQ definitions, and QR definitions are not required.
- More than one QMGR can advertise the same Q name. (work load balancing / failover)
Message Queue Interface (Procedural)

- 13 verbs (original) + 12 introduced in WMQ v7.0
- 7 are used most commonly
- Important to understand use of call parameters
# Common MQI Calls

<table>
<thead>
<tr>
<th>MQI Call</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MQCONN</td>
<td>Connect to Queue Manager</td>
</tr>
<tr>
<td>MQOPEN</td>
<td>Open a queue</td>
</tr>
<tr>
<td>MQPUT</td>
<td>Put message to queue</td>
</tr>
<tr>
<td>MQGET</td>
<td>Get message from queue</td>
</tr>
<tr>
<td>MQCLOSE</td>
<td>Close a queue</td>
</tr>
<tr>
<td>MQDISC</td>
<td>Disconnect from Queue Manager</td>
</tr>
<tr>
<td>MQPUT1</td>
<td>Put one message on a queue</td>
</tr>
</tbody>
</table>
Specialized MQI Calls

- **MQBEGIN**  Signals start of Unit of Work
- **MQCMIT**  Commits Unit of Work
- **MQBACK**  Rollback Unit of Work
- **MQINQ**  Inquire on MQ object
- **MQSET**  Set queue attributes
- **MQCONNX**  Connect with special options
## New MQI Calls in WMQ v7.0

- **MQCB** - Defines a callback function
- **MQCTL** - Start/stop message delivery
- **MQSUB** - Registers a subscription
- **MQSUBRQ** - Request services from a subscription
- **MQSTAT** - Obtain information about previous Async puts
New MQI Calls in WMQ v7.0

- MQSETMP - Set a message property
- MQINQMP - Inquire on a message property
- MQDLTLM - Delete a message property
- MQCRTMH - Create a message handle
- MQDLTMH - Delete a message handle
- MQMHBUF - Converts buffer into message handle
- MQBUFMH - Converts a message handle into a buffer
## Summary of MQ API Verbs

<table>
<thead>
<tr>
<th>Connection</th>
<th>Resource Use</th>
<th>Messages</th>
<th>Object attributes</th>
<th>Transactions</th>
<th>Message Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>MQCONN</td>
<td>MQOPEN</td>
<td>MQPUT</td>
<td>MQINQ</td>
<td>MQBEGIN</td>
<td>MQCRTMH</td>
</tr>
<tr>
<td>MQCONNX</td>
<td>MQSUB</td>
<td>MQPUT1</td>
<td>MQSET</td>
<td>MQCMIT</td>
<td>MQCLTMH</td>
</tr>
<tr>
<td>MQCTL</td>
<td>MQSUBRQ</td>
<td>MQGET</td>
<td></td>
<td>MQBACK</td>
<td>MQSETMP</td>
</tr>
<tr>
<td>MQDISC</td>
<td>MQCLOSE</td>
<td>MQCB</td>
<td></td>
<td></td>
<td>MQINQMP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MQDLTMP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MQMHDUF/MQBUF-MH</td>
</tr>
</tbody>
</table>
Application/MQ Relationship

Non-trusted

- Application
  - MQI Stub
  - IPCC
- Execution Controller
  - IPCC
- LQM Agent
  - IPCC

Queue Manager
Shared Resources

MQCONN MQDISC
MQOPEN MQCLOSE MQPUT MQGET etc.
Application/MQ Relationship

Trusted

Application

MQI Stub

LQM Agent

MQOPEN

MQCLOSE

MQPUT

MQGET

etc.

Execution Controller

MQCONNX

MQDISC

Queue Manager

Shared Resources
PCF Messages

- Specialized Message Structure, similar to that used in MQ Event messages.
- Header and multiple parameters
- Has similar functions to RUNMQSC commands
- Permits Remote Management