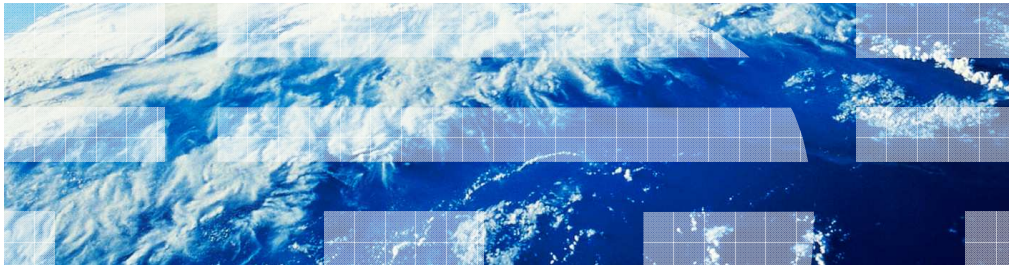# WebSphere MQ V7.0.1

Overview

This unit explains the multi-instance queue manger feature introduced in WebSphere® MQ version 7.0.1.

This unit assumes a reasonable understanding of how WebSphere MQ works.

Unit objectives

After you complete this unit, you should:

- Know the main new features of MQ version 7.0.1

After you complete this unit, you should have some understanding of the new features introduced in MQ version 7.0.1 and their purposes.

Recap - IBM WebSphere MQ V7.0

- Enhanced JMS
- Enhanced Publish-and-subscribe
- Extended verbs and behaviors for MQI programming interface
- Enhanced MQ clients for increased throughput resilience and availability
- Service Packs 7.0.0.1 and 7.0.0.2
  – Service Definition Wizard -describing MQ applications in WSDL for better governance
  – Standalone MQ Explorer download (SupportPac MS0T)
  – XA-aware API Exit
  – Continued performance enhancements
- SupportPac updates
- Evolutionary – if you know V6, you will know V7

3   Overview   © 2010 IBM Corporation

Here is a recap of the content of WebSphere MQ V7.0 that was released in 2008.

As more applications are being developed to use the standardized JMS APIs MQ's support for JMS functions have been integrated right into the heart of WebSphere MQ, giving improved usability and performance for JMS users. At the same time all the JMS features have been made available to more traditional applications using the MA API interfaces.

The publish/subscribe capabilities in MQ have been much more closely integrated with the core product and made available for the first time on the z/OS® platform

New MQ API verbs are introduced to allow access to new features, in particular asynchronous consumption of messages.

The support for MQ clients connected over TCP/IP was also revamped, increasing throughput and resilience.

Service packs for MQ 7 have already been released bringing significant functional and performance enhancement. These include a wizard to build WSDL definitions that describe MQ applications, MQ explorer delivered as a stand-alone application and improvements in the API exit functionality. Support packs are also updated to support MQ V7.

However all these changes were evolutionary – not revolutionary. If you know MQ V6 then you will know V7 too!

WebSphere MQ V7.0.1 rationale

- Objectives
  - Early delivery of non-disruptive function
  - Providing platform for other products to build on
  - Minimize migration costs
- Major themes
  - Enhanced availability options on Distributed platforms
  - Constraint relief on z/OS
  - Ongoing performance, usability and serviceability enhancements
  - Keeping pace with industry evolution in areas such as platforms and SSL

Overview

WebSphere MQ V7.0.1 is designed to be a vehicle for delivery of function earlier than might have been expected with a traditional release cycle (for example with V7.1 or V8). Some of the functions included in this release are a prerequisite on which other products such as Message Broker will build.

The major themes are enhancing the availability of the MQ server on distributed platforms, ongoing performance, usability, and serviceability enhancement including z/OS constraint relief and keeping pace with industry and standards changes.

WebSphere MQ V7.0.1 content summary

New Feature	Benefits	Details
Multi-Instance queue managers	Increases availability Can help ease system maintenance	Enables automatic failover to a standby Queue Manager instance in the event of an incident or planned outage
Automatic client reconnect	Increases availability Simplifies programming	Provides Client-connected applications with automatic detection of failures and reconnects to alternative Queue Managers
Enhanced governance	Increases visibility of changes Enables SOA Governance	Emits events whenever configuration changes are made or commands are run Service Definition wizard generates WSDL describing MQ apps
Enhanced SSL security	Simplifies security certificate management	Supports certificate checks with Online Certificate Status Protocol (OCSP) and to Certificate Revocation Lists (CRL)
Enhanced .NET support	Increases ease-of-use for .NET developers	Provides IBM Message Service Client for .NET developers Supports use of WebSphere MQ as custom channel within Windows Communication Foundation
Increased 64-bit z/OS exploitation	Increased use of z/OS system resources Provides constraint relief for virtual storage	Extends use of 64-bit storage by Queue Manager enabling more capacity such as number of open queues
z/OS log compression	Increased use of z/OS system resources Increased log performance and bandwidth	Compresses message logs produced by persistent messages
z/OS group units of work	Increased resilience	Enables Units of Work to be owned collectively by Queue Sharing Groups so that any Queue Manager in the group can process two-phase transactions from clients
Publish/subscribe interfaces	Additional control of pub/sub behavior Simplified integration for Message Broker	Exit point to dynamically modify routing and content Tools to migrate pub/sub state from MB to MQ

5   Overview   © 2010 IBM Corporation

This chart contains a table summarizing the high points of the release.

Multi-Instance queue managers and the automatic client reconnect feature together address important availability improvements. Using the two features together allows client applications (MQ API or JMS API) to connect to one instance of a queue manager and in the event of a failure of that instance to be reconnected to a new instance of the queue manager that had previously been running in standby.

Enhanced governance includes the service definition wizard and the ability to monitor configuration changes with events.

SSL security is enhanced with the support of the OCSP protocol for certificate revocation.

Dot Net client is improved and support is added for an MQ custom channel in Microsoft® Windows® Communication Foundation.

For z/OS users, more MQ resources are placed in 64-bit storage. An option is made available to compress the logs and Queue Sharing Group unit of work management is improved.

The publish/subscribe interfaces are also extended with additional exits and controls which will be used by WebSphere Message Broker.

Installation and delivery

- WebSphere MQ V7.0.1 is a modification release on the V7 base
  - Which means limited scope for new objects/attributes
    - Minimizes migration aspects
- On distributed platforms, it is available in two ways
  - A fix pack for upgrade from existing V7 installations (which can be backed out)
  - A replacement V7 installation image
  - Customers ordering V7 will now get V7.0.1
    - Single service stream for V7.0
- On z/OS, it is available as a modification level release
  - Migration supported from V6 and from V7.0.0
  - Customers ordering V7 will now get V7.0.1
  - New ZPARM option to control whether new function is available

6   Overview   © 2010 IBM Corporation

Because MQ 7.0.1 is a modification release over MQ 7.0.0 the base structures used by the product are unchanged, which means that migration issues, from MQ 7.0.0, are minimized.

On distributed platforms it is available as a reversible upgrade. However new orders for MQ7 will be delivered as MQ 7.0.1.

On z/OS a similar situation holds except that the z/OS function does include some features that are not compatible with earlier releases, for example log compression. For this reason a new ZPARM feature, or switch, is implemented. This switch is initially in the OFF position which disables the MQ 7.0.1 only features. The features can be enabled at a later date.

Distributed platforms: Multi-instance queue managers

- Basic failover support without HA coordinator
  - Faster takeover: fewer moving parts
  - Cheaper: no specialized software or administration skills needed
  - Windows, UNIX®, Linux® platforms
- Queue manager data is held in networked storage
  - NAS, NFS, GPFS™ and so on so more than one machine sees the queue manager data
  - Improves storage management options: formal support for these even without failover configuration
- Multiple (2) instances of a queue manager on different machines
  - One is "active" instance; other is "standby" instance
- Instances share data, so it is the SAME queue manager

7   Overview   © 2010 IBM Corporation

This new feature introduces "Basic failover" provided by MQ. There is no interaction with other resources like disks, IP addresses, databases, user applications and so on. There is also no sophisticated control over where the queue managers run and move to (for example a 3-node HACMP™ setup).

Architecturally, this is a simple solution with the data shared between systems. For the first time this gives formal support for placing the MQ queue and log data in other than local storage.

Just as with other high availability configurations the takeover is in essence a restart of the queue manager, so non-persistent messages are discarded, queue manager channels go into retry and so on.

It is possible to have instances of the same queue manager running on different machines. The queue manager can be defined on many machines but can only be running on at most two - one active and one standby.

The two instances are not different queue managers – they share the data – they are the SAME queue manager.

Distributed platforms: Multi-instance queue managers: Notes

- MQ is NOT becoming an HA coordinator
  – Generally, if other resources also required, use an HA coordinator such as HACMP
  – Service objects can restart applications with qmgr but limited control
  – Message Broker will integrate with and exploit this MQ function

- The IP address is not taken over
  – Channel configuration needs all possible addresses
  – CONNAME('host1(port1),host2(port2)') syntax extension on all platforms including z/OS

- New options for crtmqm/strmqm/endmqm to control operations
  – Manages active and standby instances

- Support for networked storage over modern network file system protocols
  – For example, NFS v4 (not v3)
  – Tool shipped to validate configuration

8   Overview   © 2010 IBM Corporation

MQ is not becoming a High Availability Coordinator, if you have requirements to switch over a range of other resource managers or services you should still consider a solution such as HACMP. The queue manager can start service objects during restart, but with no sequence control or automatic monitoring. It is intended that WebSphere Message Broker will be integrated to restart with its queue manager.

Because the IP address that the original active queue manager was running on is not "taken over" by the newly active queue manager, all channel definitions must be "aware" of all possible IP addresses they might connect to. This is addressed by extending the CONNAME parameter to accept a list of IP addresses.

New options are supplied on the create, start, and end commands to allow the management of active and standby instances.

Support for the type of shared storage is limited by the degree of POSIX compliance the shared storage supports. Modern file systems like Network File System (NFS) version 4 and the Common Internet File System (CIFS) are supported but older systems may not be. A tool which tests compliance is shipped with MQ.

Automatic client reconnection

- The MQ Client Code attempts reconnection to Qmgr after failure
  - Successful reconnect is hidden from application code
  - Re-opens queues and other qmgr objects, re-establishes subscriptions
- Uses the list of addresses in CONNAME to find queue manager
  - MQSERVER environment variable also understands list
    - MQSERVER=SYSTEM.DEF.SVRCONN/TCP/host1(1414),host2(1414)
- Can reconnect to the same or different Queue Manager
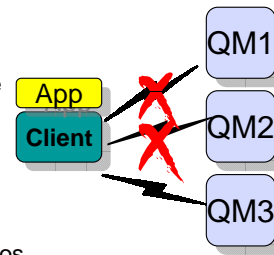
The MQ client code has been extended to allow it to attempt to reconnect to a queue manager in the event that the connection is lost. If the client successfully reconnects then all the resources that were originally open will be restored so far as possible by the client code to allow the application to continue.

The Client code uses the list of IP addresses in the extended CONNAME field to determine the list of possible locations for queue managers to switch to.

The client can either switch to any available queue manager, or to only the same queue manager it was previously connected to. It is important to understand here that this would include another instance of the same queue manager. Automatic client reconnection together with multi instance queue manger can allow an application program to be "seamlessly" reconnected in the event of a queue manager failure and takeover.

Publish/subscribe enhancements

- Option to discover if no subscribers (user or proxy) during MQPUT/PUT1
  – MQPMO_WARN_IF_NO_SUBS_MATCHED
  – MQRC_NO_SUBS_MATCHED

- Will guarantee that no one has received the publication
  - But does NOT guarantee that anyone will definitely receive the publication

- Publish exit
  – When a publication is made, this exit is invoked for each valid subscriber
    • Runs "inside" the queue manager
  – Can change routing destination, contents of message, contents of message, Can inhibit publication

The publish/subscribe APIs have been extended with two major additional features.

The "No Match" option: If requested a put to a topic will give a warning return code if no subscribers qualify for the message. Note: this return code guarantees that no subscriber received the message however it is possible that return code zero is returned but no subscriber actually gets the message.

The "Publish Exit" allows an exit to be run for each subscriber. One use of this would be to examine the contents of the message and use this to determine whether to deliver the message.

XMS and WCF

- XMS .Net classes
  - Now shipped as part of the product instead of SupportPac
  - Internally, it exploits V7 API, in same way as JMS implementation

- Windows Communication Framework (WCF)
  - Underpins .NET Web services and Messaging
  - Built-in Transports – for example: MSMQ, HTTP, HTTPS, Named Pipes, TCP/IP, and so on.
  - MQ transport is added as a "custom channels"

- WCF support now included in product
  - Previously an alphaWorks prototype
  - Built on XMS .Net classes

XMS is a set of APIs that provide an interface to MQ for .NET users that is closely modeled on JMS. In MQ V6 this was a support pack, but is now fully integrated and is implemented directly using the same internal APIs as the JMS interface.

Windows Communication Framework is a Microsoft High level interface for invoking Web services and messaging. It comes with some built-in transports, but is designed to be extensible. The WCF support in MQ 7.0.1 adds MQ transport as a WCF "custom Channel".

In earlier releases an "IBM AlphaWorks" project provided similar support, this is now fully "productized" and integrated into MQ 7.0.1 and is built on top of the MQ XMS APIs.

Enhanced governance

- Command and configuration events for distributed platforms
  - Matching already-available z/OS function
  - Successful commands – MQSC or PCF – recorded as event messages
  - Configuration changes report "before" and "after" definitions of objects
  - Provides a record of who changed what and when

```
display qmgr event
AMQ8408: Display Queue Manager details.
    QMNAME(V7) CMDEV(ENABLED) CONFIGEV(ENABLED) …
```

- SSL OCSP support
  - Now commonly used as alternative to LDAP-based CRLs
  - Simpler to manage as no need to have an LDAP server
  - Can use details provided in inbound certificate

12   Overview   © 2010 IBM Corporation

Command and configuration "events" can now be written to queues to show the before and after status after a successful change command giving a record of "who" did "what". This matches the already available function in z/OS and the formats of the events are common across z/OS and distributed platforms.

Many monitoring tools will already understand the formats; they may just have to be pointed at the appropriate EVENT queue.

Remember that the event queues can be redefined as aliases to topics, so events can be delivered to multiple consumers using pub/sub.

OCSP is a standard that allows servers to determine the validity of a security certificate by requesting status information from a server. It can be used as an alternative or in addition to the existing LDAP based Certificate Revocation Lists.

z/OS specific changes

- In V7.0.1 more Queue Manager storage moves to 64-bit
  - 64-bit Queue Indices, 64-bit Lock Manager, 64-bit Security Manager
- z/OS Log Compression
  - Can increase the throughput possible for persistent messages
  - May reduce the size of your logs
  - SMF 115 records updated to show compression rates achieved
  - Controlled by zPARM option at queue manager level.
- Group-level Units of Recovery for z/OS QSG
  - Requires use of the Extended Transactional Client – example from WebSphere Application Server
  - Two-phase/global transaction can now be owned by a Queue Sharing Group instead of by individual queue managers
    - These in-doubt transactions can be resolved on **any** QMGR in the QSG

Some items in MQ 7.0.1 are specific to the z/OS version. MQ 7.0.1 increases the number of objects that can be placed in 64 bit storage thereby increasing the number of objects like messages on indexed queues that can be handled by a queue manager.

Two specific items of new function are "Log Compression" and "Group Level Units of Recovery."

Log compression applies run length encoding compression to log records before they are written to disk. In appropriate circumstances the compression may increase throughput by reducing the IO volume on the log. The option is switched on using a new "zPARM" option. SMF records are written that indicate the degree of compression that is being achieved.

Group Level Units of recovery apply to updates made with the extended transactional client to objects in queue sharing groups. The unit of work is "owned" by the QSG rather than by an individual queue manager, this means that in-doubt transaction resolution can be performed by any queue manager in the QSG rather than only by the queue manager that initiated the transaction.

Unit objectives

After you complete this unit, you should:

- Understand the new features of MQ version 7.0.1.

- Especially multi-instance queue managers and reconnecting client, .NET support and the z/OS enhancements.

Overview   © 2010 IBM Corporation

Now that you have completed this unit you should have some understanding of the new features introduced into MQ 7.0.1.

The multi instance queue manager together with the reconnecting client gives a software based high availability solution. Additional features for Windows and z/Os extend the platform features available to MQ.

This unit does not attempt to cover the full range of syntax and options available, for which you should refer to the product information center.

Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?

- Did it help you solve a problem or answer a question?

- Do you have suggestions for improvements?

Click to send e-mail feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_iea_701_100_overview.ppt

This module is also available in PDF format at: ../iea_701_100_overview.pdf

15   Overview   © 2010 IBM Corporation

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

16   © 2010 IBM Corporation