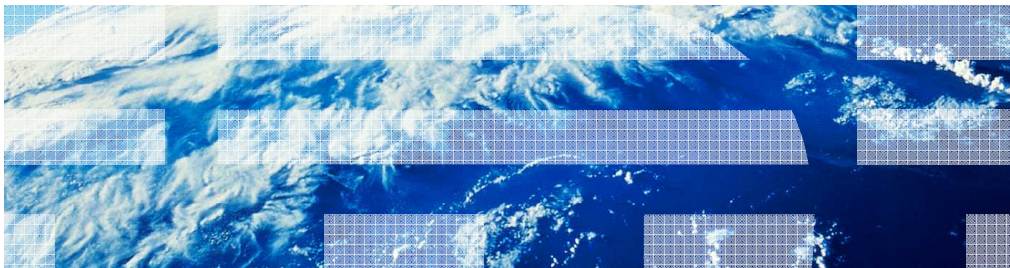


---

# WebSphere Message Broker Version 7

Publish / subscribe architecture and applications



This session describes the changes that have been made to the publish / subscribe function in WebSphere Message Broker version 7. This area has been changed significantly in this version, and now makes use of the publish / subscribe functions in WebSphere MQ version 7.

## Table of contents

- High level publish / subscribe introduction
- Current publish / subscribe implementation in version 6
- Exploitation of WebSphere MQ version 7.0.1 within Message Broker version 7
  - Ensuring that previous function / capability continues to work
    - Publication node interfacing with MQ Matchspace
    - Content-based filtering
    - Publish exit
  - Additional publish / subscribe enhancements within Message Broker
    - 'Out' and 'no match' terminals on publication node
- Migration options
  - State migration tool

This session will first provide a high-level overview of publish / subscribe applications, and how this compares to point to point application connectivity. The background of the WebSphere MQ and Message Broker publish / subscribe functions are explained, and how this has now been implemented in version 7.

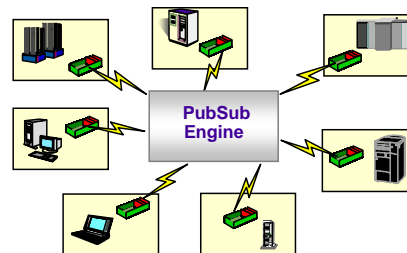
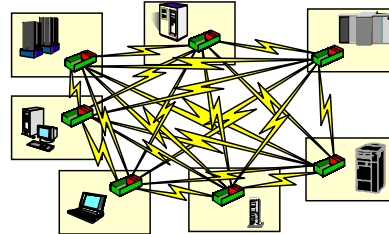
Finally, migration of publish / subscribe applications will be covered. This will include migration of the publication and subscription state from Message Broker to MQ.

This session will first provide a high-level overview of publish / subscribe applications, and how this compares to point to point application connectivity. The background of the WebSphere MQ and Message Broker publish / subscribe functions are explained, and how this has now been implemented in version 7.

Finally, migration of publish / subscribe applications will be covered. This will include migration of the publication and subscription state from Message Broker to MQ.

## Publish / subscribe introduction

- Point-to-Point is not always suitable
  - Large number of senders or receivers
  - Senders and receivers frequently change
    - Senders need to know exactly who to send messages to
    - When a change occurs, clients need to be updated
    - Can become unmanageable
- Solution
  - Manage lists of senders and receivers in one place
  - Allow applications to **publish** messages
  - ...and other applications to **subscribe** to these messages



3

Publish/subscribe

© 2010 IBM Corporation

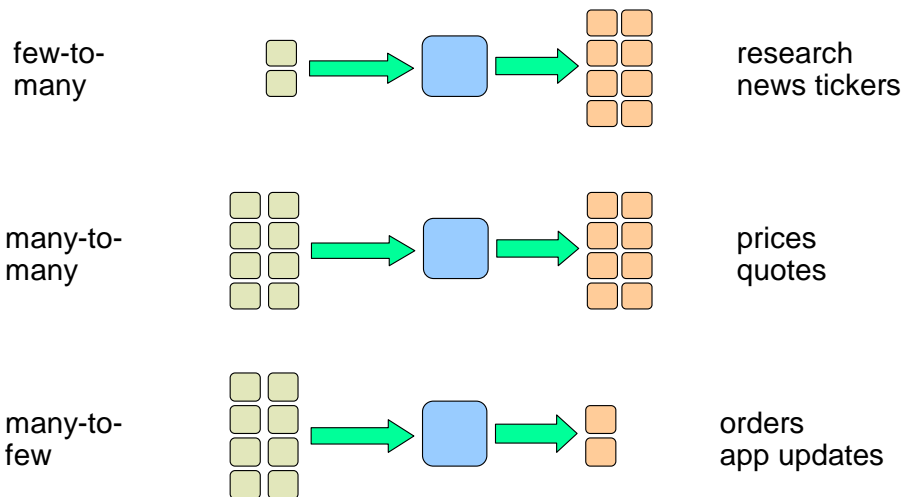
The publish / subscribe architecture can be used in scenarios where a point to point architecture might not be suitable. For example, if there are a large number of senders or receivers, then it can be difficult to define these end-points. Senders, or sender applications, generally need to know who they are sending messages to. If the senders and receivers change frequently, then keeping this information up to date for routing can be difficult, and can become unmanageable.

These requirements are best addressed with a publish / subscribe architecture.

Information can be published using topic-based delivery, and applications publish messages on a given topic. The receiving applications subscribe to messages on a given topic, and the publish / subscribe matching engine ensures that messages are delivered to those applications that request it. A topic is a defined identifier for the particular information that the subscriber is sending to the receiver.

Publishers and subscribers are not known to each other. A publisher will not know who has subscribed to a particular topic, or if there are any subscribers. Similarly, a subscriber will not know which application or publisher has provided information on the topic he has subscribed to.

## Information delivery models

4  
4

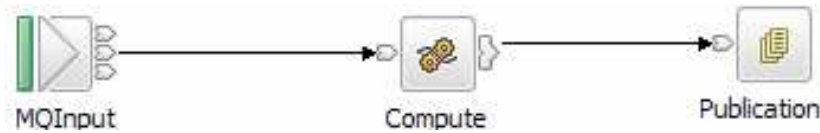
Publish/subscribe

© 2010 IBM Corporation

The publish / subscribe model can be used in different ways. The number of publishers might be small, and the number of subscribers might be large. An example of this might be a large sporting event, where a large number of individuals want to receive information about a small number of events.

Conversely, the number of publishers might be large, with a small number of subscribers. An example of this might be an internet ordering system, where a large number of clients place orders, but the subscriber is a single instance of the order application.

## Example of a message flow that publishes messages



- Publication messages handled within the flow
  - Publication node receives publication messages and distributes to the subscribers matching the topic within the publication message
- MQ based subscription messages registered using a special queue
  - SYSTEM.BROKER.CONTROL.QUEUE
  - Do not need to define a queue to handle subscription requests

This slide shows an example of a message flow that is publishing messages. Input to the flow is provided by a standard MQ input node. This is passed to a compute node for the particular processing that is required. In particular, the compute node is responsible for specifying the topic that the publication node is going to use. This topic is contained within the header of the published message. The message is then to a publication node, which passes the request to the publication engine. This uses the topic to determine which subscribers should receive this message.

The message flow is not used to manage the subscribers to this topic. The subscribers to a topic have previously registered their subscriptions, and this information is stored on a queue, with the name as shown on this slide.

## Publish / subscribe in Message Broker version 6.1

- There are currently two separate publish / subscribe engines
  - MQ publish / subscribe
  - Message Broker publish / subscribe
- Different administration and configuration options and tools
- Different security model
- Different capability

To set the context for Message Broker version 7, it is useful to understand the earlier implementations of publish / subscribe.

First, in WebSphere MQ, publish / subscribe was first introduced as a Support Pack, then later put into the main product in MQ version 6. This function provides topic-based publish/subscribe only.

Secondly, Message Broker also contains an implementation of publish/subscribe. This function provides both topic-based and content-based publish/subscribe functions.

These two implementations can not both be active on the same queue manager, and the implementation within Message Broker does not take advantage of the implementation within MQ.

If you are using Message Broker version 6 or 6.1, you must use the publish/subscribe function provided by Message Broker.

## Publish / subscribe in WebSphere MQ version 7

Publish / subscribe function	MQ version 7.0.1	Message Broker 6.0/6.1
Topic based matching	Yes	Yes
Retained publication support	Yes	Yes
RFH (1) Support	Yes	Yes
RFH 2 Support	Yes	Yes
Security / Access Control (ACLs)	Yes	Yes
Centralized graphical configuration	Yes	Yes
High availability option	Yes	Yes
High performance non-persistent option	Yes	Yes
Available on all platforms	Yes	Yes

7

Publish/subscribe

© 2010 IBM Corporation

WebSphere MQ version 7 made extensive enhancements to its publish / subscribe function.

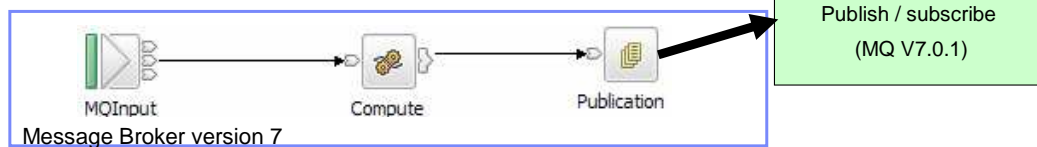
MQ version 7 now has a publish / subscribe capability that provides similar capability to Message Broker version 6, for topic-based scenarios.

This development leads naturally to the evolution of publish / subscribe capability in Message Broker. Message Broker version 7 has been changed to take advantage of the publish / subscribe capability in WebSphere MQ version 7. All topic-based publish / subscribe management is now managed by WebSphere MQ. This provides a much clearer implementation for publish / subscribe, and allows Message Broker to complement WebSphere MQ, with the addition of content-based publish / subscribe.

In Message Broker version 7, publish / subscribe administration will be done using the MQ Explorer. Operating system levels are the same as MQ wherever possible and product prerequisites are the same.

This also means that the “username server” that was required for Message Broker publish / subscribe can be removed.

## How Message Broker uses MQ publish / subscribe



- Publication node interfaces with the matching engine within MQ
  - Sends publication messages to MQ
    - Parses message being propagated to Publication node
    - Calls MQ API to allow MQ to match and deliver to subscribers
  - The publication node will continue to have the same external behaviour
    - Supports RFH, RFH2, and message properties
    - Will continue to support subscription points for RFH2 messages
- All internal and external publish / subscribe components utilize the MQ publish / subscribe function
  - Accounting and statistics reporting
  - Notification messages (such as flow started, stopped)

8

Publish/subscribe

© 2010 IBM Corporation

The primary interface to the publish / subscribe function from within a message flow is with the publication node.

This has been changed in Message Broker version to utilize the updated publish/subscribe engine in WebSphere MQ version 7.0.1. Message Broker version 7 therefore has a prerequisite level of WebSphere MQ version 7.0.1.

Although the internal mechanisms have changed, this does not effect the external interfaces.

The RFH1 and RFH2, streams, subscription points, all work as in previous versions.

Message Broker also uses publish / subscribe for other functions, such as for Accounting and Statistics, the Monitoring function, and for message communication between internal product components. These have also been updated to ensure that they correctly utilize the new publication engine.



## Content-based filtering

- Content based filtering allows a subscriber to restrict the messages that it wants to receive
  - When a client registers a subscription, it can specify a filter to be applied to the content of fields within each publication message
  - The filter itself is entered as an SQL expression
    - Body.Name LIKE 'Smit%'
- Message Broker supports filtering on the whole message, including the body of the message
- The MQ version 7.0.1 matching engine supports filtering on the header (Message Properties) of the message
- MQ queue manager will use message body parsing provided by Message Broker to support content-based filtering
- Message Broker content-based filtering is run in nominated execution groups

Content based filtering allows a subscriber to restrict or filter messages that it wants to receive, in addition to the topics that it specified.

This filter is specified as an SQL expression.

WebSphere MQ version 7 supports a limited amount of filtering on the message header and message properties, but can not filter on the body of the message. This is because it can not parse or understand this part of the message.

Message Broker version 6 and 6.1 provides content-based filtering on the body of the message , and this has been maintained in version 7.

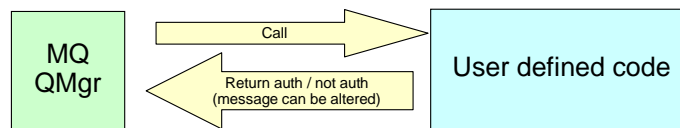
Message Broker and MQ work together to provide content-based filtering on the whole message. This uses the parsing and filtering capability provided by Message Broker, and the publish/subscribe function of MQ.

This Message Broker function is located within an execution group, and you can nominate the execution group you want this to run in.

If at least one execution group within a broker has been enabled to support content-based filtering, the associated queue manager will support that function. Each execution must have all the required application artifacts, such as the message definitions, to enable content-based filtering.

## Publish exit

- In Message Broker, you can implement your own exit to handle authentication and authorization
- Allows you to inhibit the message from being sent to the subscriber
- Allows message to be changed / updated
  - Exposes MQMD, headers and payload
  - Change destination of message
  - Change body of message
- Configured using ini file
- API is different from MQ or Message Broker, so a porting exercise is required



10

Publish/subscribe

© 2010 IBM Corporation

Both WebSphere Message Broker and WebSphere MQ version 6 support an exit in which you can deploy additional application of code. This can be used to make a decision on whether a message should be delivered to a specific subscriber based on the subscriber details and the message content.

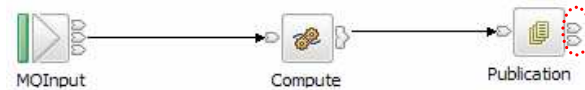
When a message is published, the exit is called just before sending to subscriber, and can be used to determine final routing decisions. The message can be changed by the exit code if required.

This exit is closely coupled to publish / subscribe function, and in version 7, this function is now available in WebSphere MQ. The exit has also been enhanced in version 7, and now allows you to change most components of the published message.

The programming interface has been changed from earlier versions, so if you use this function, you should review this code. This interface is provided in the C programming language.

## No-match and Out terminals

- Terminals provided on publication node



- Drives flow depending on subscribers matching publication
  - No-match terminal
    - If there are no matching subscribers
  - Out terminal
    - If one or more subscribers match

In Message Broker version 6, the publication node has just a single terminal, the input terminal.

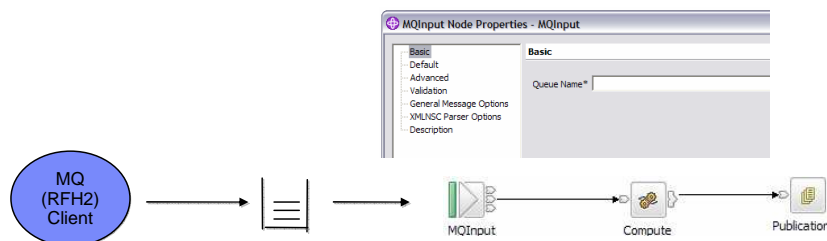
In version 7, the publication node has been enhanced to allow you to detect if no subscribers exist for the publication. This is done using the two new terminals on the publication node.

First, the “no-match” terminal. The publication message is propagated down this terminal if no subscribers match the publication message.

Second, the “out” terminal. The publication message will be propagated down this terminal if at least one subscriber matched this publication message.

## Using the MQ Input node for publish / subscribe

- Message Broker is queue based



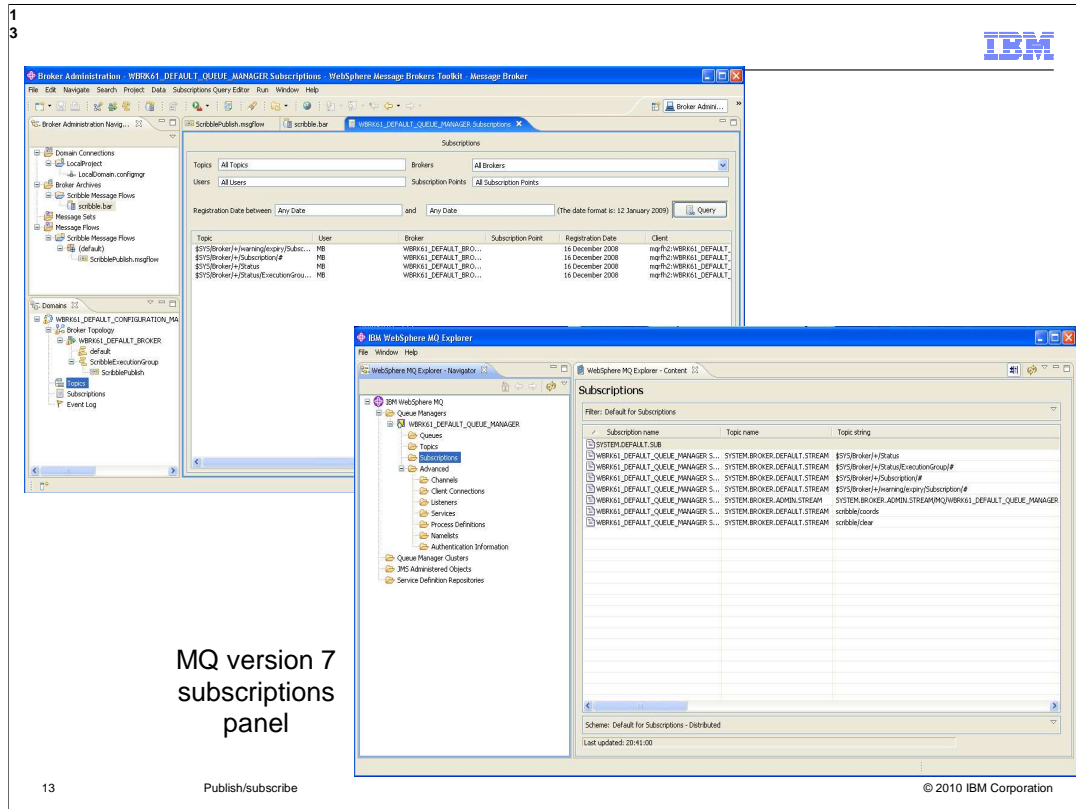
- Receives its input from a queue (containing an RFH message header)
- This is still supported within Message Broker version 7.0

This is a typical Message Broker flow for handling publish / subscribe using the MQ transport.

The MQ Input node is configured to read a message from a queue. This message can be in RFH or RFH2 format.

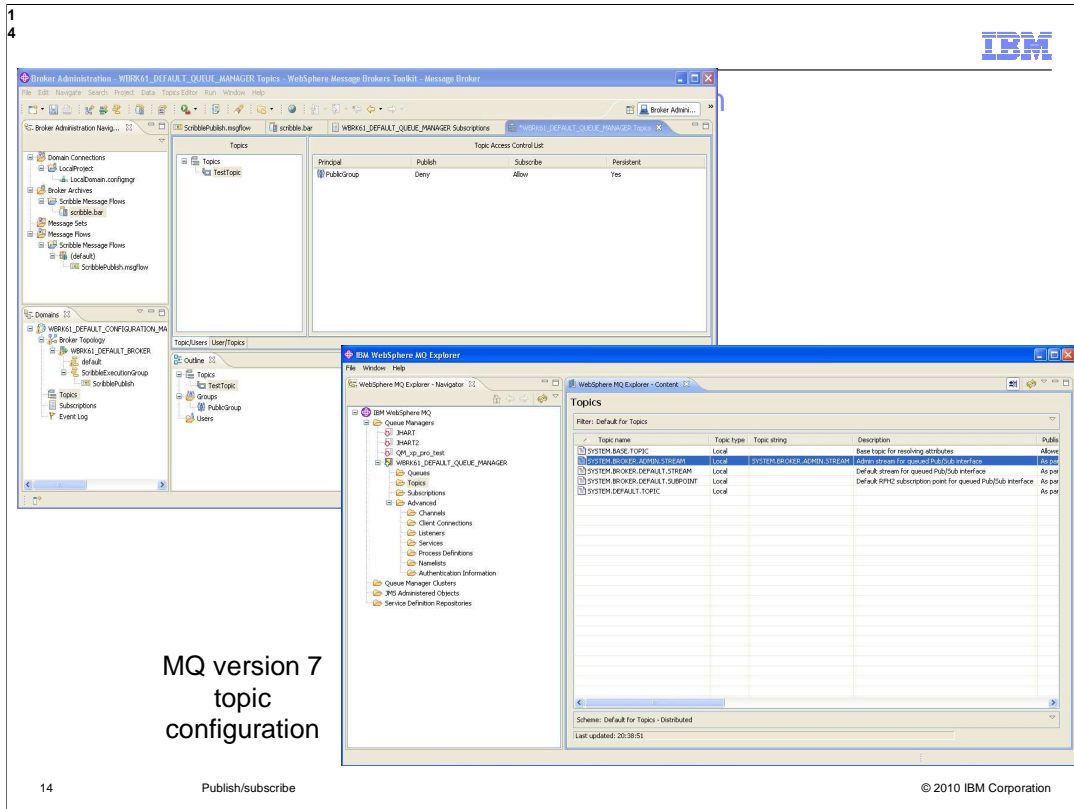
The Publication node is responsible for processing the publication messages.

This application is fully supported in Message Broker version 7.



Since the publish / subscribe function has moved from Message Broker to MQ, the administration of the publish/subscribe system is now done using MQ facilities.

For example, the management of subscriptions is now done using the MQ Explorer. The screen captures shown here show subscription management in Message Broker version 6.1, and how this is now implemented in MQ Explorer in MQ version 7.



Similarly, these screen captures show the topic management and access control administration function in Message Broker version 6.1, and the equivalent function in MQ Explorer in MQ version 7.

## Supported migration paths

- Migration from Event Broker to WebSphere MQ version 7.0.1
  - Event Broker is a subset of Message Broker
    - Code stabilized at Event Broker version 6.0
  - Requires Message Broker to support content-based filtering
- Migration from Event Broker to Message Broker
  - Continues to support content-based filtering
- Migration to Message Broker version 7
  - Supported versions
    - Message Broker version 6.1
    - Message Broker version 6.0
- Use standard Message Broker version 7 migration tools
  - Plus publish / subscribe state migration tool

When migrating a publish / subscribe application, you will have several choices, depending on which implementation of publish/subscribe you have today.

First, if you have WebSphere Event Broker today, then the publish/subscribe application is likely to be topic-based, since Event Broker does not provide tools to enable content-based filtering. In this instance, you should consider a migration to WebSphere MQ version 7. Note that Event Broker was stabilized at version 6.0. However, a direct migration to WebSphere MQ version 7.0.1 is supported.

Second, if you do have a requirement for content-based filtering, or if you use Message Broker today, then a migration to Message Broker version 7 is more appropriate.

The majority of the migration to version 7 will use the standard migration tools. Migration to version 7 is fully described in the IBM Education Assistant session covering migration. For publish / subscribe scenarios, an additional step is required to migration existing subscriptions and retained publications. This is described on the next slide.

## State migration tool (1 of 2)

- Allows the publish / subscribe state held within Message Broker to be migrated to WebSphere MQ
  - Subscriptions
  - Retained publications
  - ACLs
    - Generates script to migrate 'positive' ACLs
- Command : "migmbbrk"
  - Shipped in WebSphere MQ version 7.0.1
- Required when using any aspect of publish/subscribe
  - Directly exploiting publish / subscribe with publication node
  - Accounting and statistics
  - Monitoring

The state migration tool is provided in WebSphere MQ version 7.0.1. This tool takes existing publish / subscribe state information such as subscriptions and retained publications from Message Broker and imports it into WebSphere MQ.

In the case of access control lists, this information is not loaded directly into MQ, but is written to a script file. This is because the security model for publish / subscribe is different in MQ compared to Message Broker. For example, Message Broker supports both positive and negative ACLs, whereas MQ only supports positive ACLs, with the root topic being a negative ACL. You might therefore need to change the security approach when migrating to MQ. The generated script file contains a series of MQ authorization commands, and can be used to load into the queue manager if required.



## State migration tool (2 of 2)

- Can be run against Message Broker version 6.0 / 6.1 or Event Broker version 6.0
  - Can not be run against an Message Broker version 7 broker
  
- "migmbbrk" needs to be run twice
  - Use the "-t" option – this creates the topic objects within MQ and generates the security script
  - Use the "-c" option – this populates MQ matchspace (subscriptions and retained publications)
  
- Usage
  - migmbbrk [-r] [-t] [-c] [-o] [-s] [-z] [-l] -b  
    BrokerName
  - Produces log file to record migration status

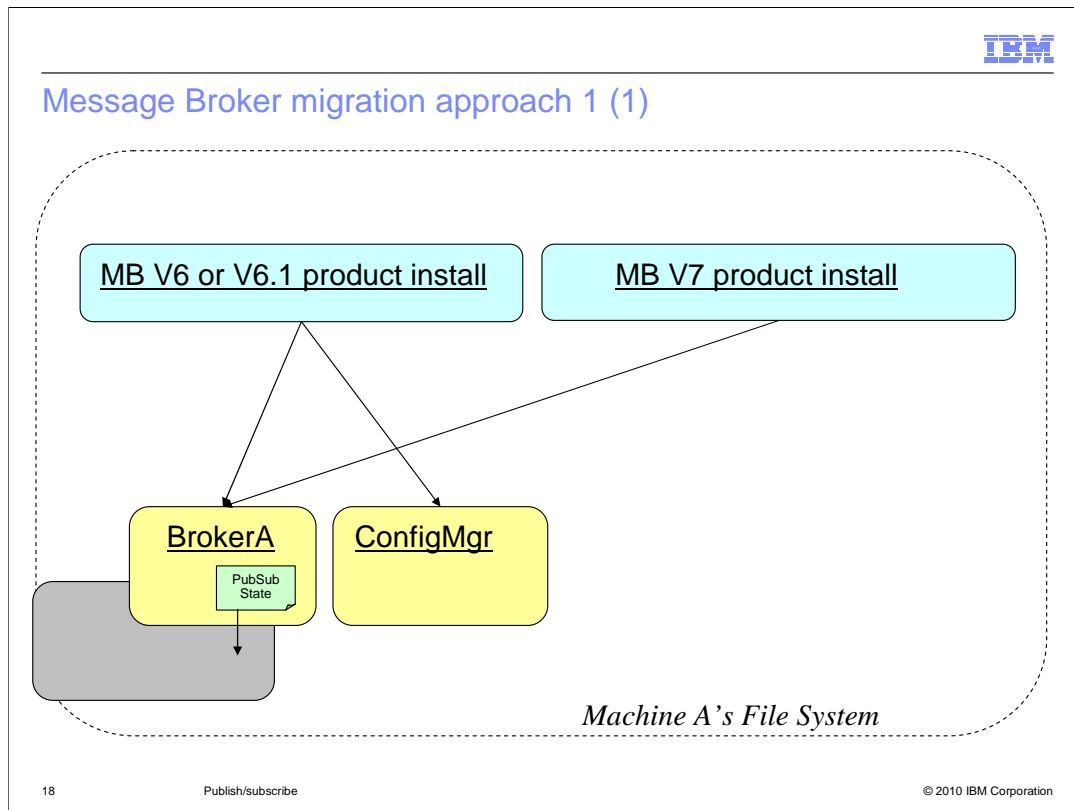
The state migration tool can only be run against Message Broker or Event Broker version 6.0 or 6.1. It can not be run against Message Broker version 7, since publish/subscribe state is no longer held in this environment.

The tool should be run twice.

The first time, you should use the "-t" option. This will associate the defined topic objects in Message Broker in the new MQ version 7.0.1 queue manager. It will also generate the script file containing the security ACL definitions.

The second time you should use the "-c" option. This will migrate the existing subscriptions and retained publications into the version 7 queue manager.

The "-r" option is a rehearsal option. This allows you try the command. The command output is available, but the command does not make the final updates into the MQ queue manager. The "-z" option forces the command to operate a subsequent time, even if it has been previously run.

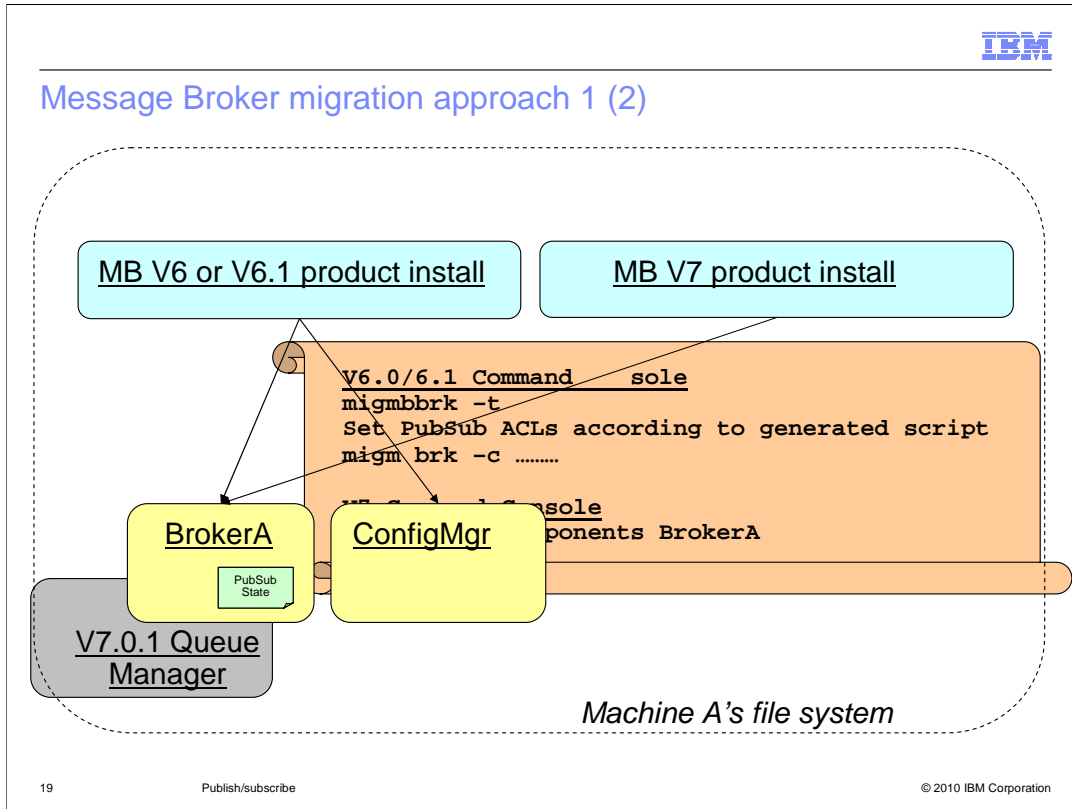


The first approach to migration is to perform this in stages. At each stage of the migration, a full working system is available for use.

This example shows the steps to migrate from Message Broker version 6.0 or 6.1 to Message Broker version 7, using this approach. The existing installation of Message Broker will have a broker and configuration manager, and the version 6 broker will contain publish/subscribe state such as subscriptions.

The first step in the migration is to ensure that the Message Broker installations have the right fixpack level to support WebSphere MQ 7.0.1. For Message Broker version 6.0, this is fixpack 10, and for Message Broker version 6.1, this is fixpack 5. Note that these are higher pre-requisites than normal, and are specifically required for publish/subscribe.

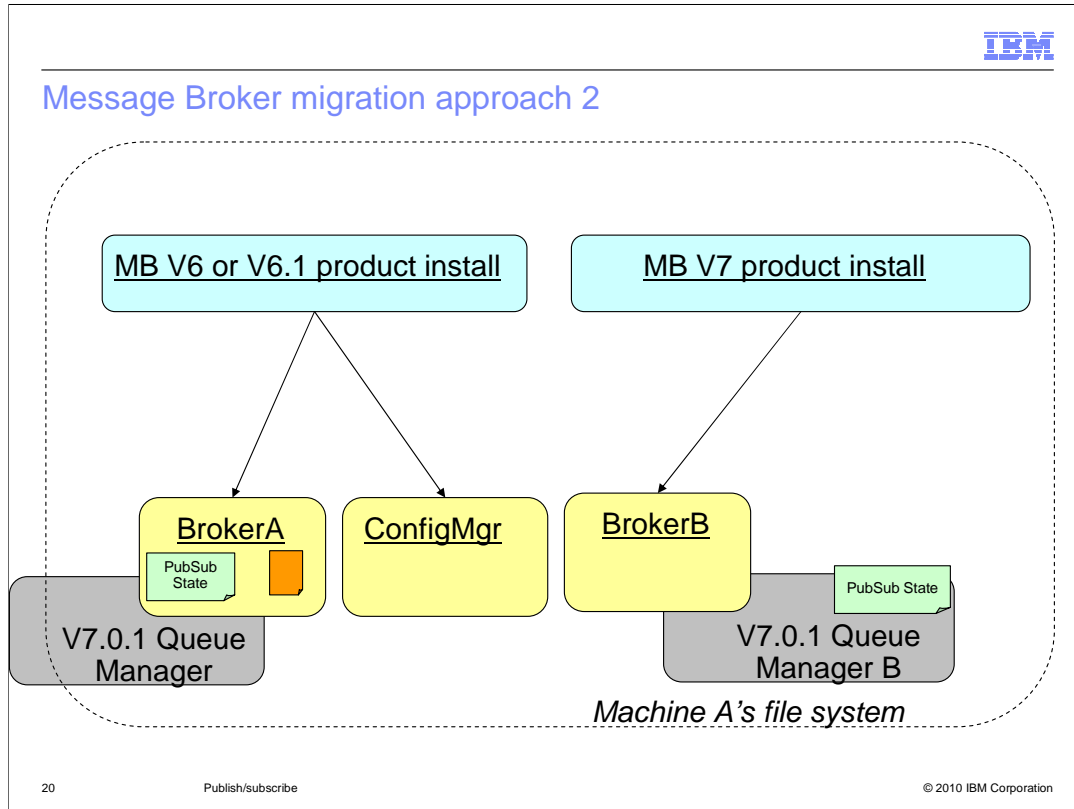
The next step in the migration is to upgrade the underlying queue manager to use WebSphere MQ version 7.0.1. Once MQ has been upgraded, Message Broker version 7 can be installed alongside the existing version 6 installation.



You are now in a position to migrate the publish/subscribe state data, by using the state migration tool, as shown.

First, use the “-t” option to migrate the topic definitions and create the ACL list. Then, use the “-c” option to migrate the subscriptions and retained publications. This command should be run against the existing Message Broker installation, using the appropriate command console.

Once this has completed successfully, you can now proceed to creating the version 7 broker, and migrating the applications using the normal migration tools.



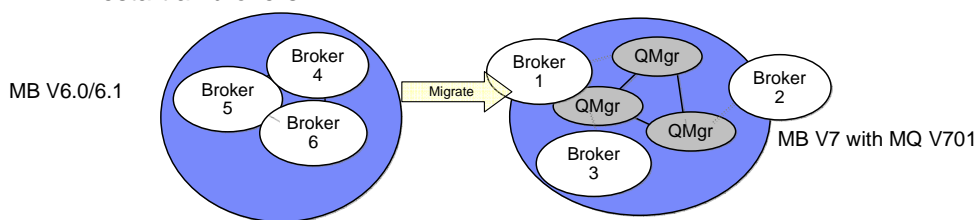
The second approach to migration is to build a completely separate environment, and run the two environments at the same time, whilst migrating components to the new system.

The starting point is the same, with the same pre-requisite levels for Message Broker and MQ version 7.0.1 installed.

In this case, the version 7 broker must be built manually, and the publish/subscribe state created from scratch.

## Collectives migration

- Message Broker allows brokers to be interconnected and logically share a single matchspace by joining brokers within a 'collective'
  - MQ V7.0.1 supports publish / subscribe clusters which uses a different protocol
- All brokers will need to be migrated together
- For each broker in the Collective perform these steps
  - Remove broker from Collective
  - Migrate broker as described in migration approach number 1
  - Add queue manager associated with Broker to publish/subscribe cluster
- Restart all brokers



21

Publish/subscribe

© 2010 IBM Corporation

Finally, a more complex publish / subscribe scenario uses collectives. Collectives are groups of brokers that communicate and allow subscribers and publishers to use different broker instances. For example, a subscriber might register a subscription using one broker, and another application will publish data in a different broker. The collective will ensure that the published data is sent to the registered subscriber.

When publish / subscribe is implemented using WebSphere MQ version 7, support for collectives is provided by using MQ clusters. The migration approach is therefore to migrate each broker version 6 instance to a corresponding MQ queue manager, and then associate the queue manager with the MQ cluster.

When performing this migration, the collective must be migrated completely, before it can be used. It is not possible to combine publish / subscribe components within the collective at different versions.

You should ensure that no messages are in flight. For each broker in the Collective, first upgrade MQ to version 7.0.1, then migrate each broker as described in migration approach number 1.

Once all of the brokers have been upgraded, run the MQ command to inform the queue managers to use the MQ publish/subscribe cluster capability.

## Summary

- Close synergy between WebSphere MQ and WebSphere Message Broker
  - MQ provides all topic-based publish / subscribe
  - Message Broker provides additional services
  
- The Message Broker publication function is removed
  - Message Broker will rely on the publication engine within WebSphere MQ version 7.0.1
  
- A migration path is provided to allow you to easily upgrade to the appropriate new versions
  - State migration tool

In summary, the publish / subscribe function in Message Broker version 7 has been enhanced to take advantage of the MQ capability. All topic-based publish / subscribe is managed using MQ, including the administration function. If you require content-based filtering, then Message Broker version 7 extends the base MQ capability.

Migration from Message Broker version 6 is fully supported, and a state migration tool enables you to migrate your existing subscriptions, retained publications and access control lists.



## 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\\_WMB7\\_GeneralFacilities\\_Publish\\_Subscribe.ppt](mailto:iea@us.ibm.com?subject=Feedback_about_WMB7_GeneralFacilities_Publish_Subscribe.ppt)

This module is also available in PDF format at: [..\\WMB7\\_GeneralFacilities\\_Publish\\_Subscribe.pdf](..\\WMB7_GeneralFacilities_Publish_Subscribe.pdf)

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



## Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, Current, and WebSphere 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 other IBM trademarks is available on the Web at "[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)" at <http://www.ibm.com/legal/copytrade.shtml>

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.

© Copyright International Business Machines Corporation 2010. All rights reserved.