Cluster setup and basic usage of clustered queues and topics in WebSphere MQ 7

Angel Rivera (rivera@us.ibm.com)
WebSphere MQ Unix® and Windows Level 2 Support
Date: 16-Jul-2013
Agenda

- Scope and audience for this presentation
- Previous WSTEs on intermediate/advanced clustering themes
- Steps to create a basic cluster
- Exploring behavior of put/get for:
  - Non-clustered queue
  - Clustered queue
- Exploring behavior of publish/subscribe
  - Non-clustered topic
  - Clustered topic
Scope and Audience

- The Scope for this presentation is very narrow: it is very practical.

- To provide step-by-step instructions for the setup of a small cluster.

- To provide concrete examples of doing a put and a get on a clustered queue and a clustered topic.

- The audience is for beginners.
Previous WSTEs

- The following are previous WSTEs that provide intermediate and advance details on MQ clusters
  - Webcast: WebSphere MQ V7 Clustering
  - This presentation gives an overview of clustering and explains how cluster works and its configuration. It also covers the clustering workload balance, clustering best practices, and also goes over the differences between clustering and queue sharing group.
  - Level of Difficulty: Intermediate
Previous WSTEs - continue

- WebSphere MQ V6 Queue Manager Clusters
  Discusses the purpose, usage, and configuration of MQ clusters in a distributed environment.

- WebSphere MQ Queue Manager Clustering
  Cover basic and intermediate MQ queue manager cluster configuration for z/OS and distributed platforms (Windows, UNIX, iSeries). It also covers common problems and problem diagnosis.
Setup of a small cluster

- For the setup of the cluster, including the creation of the queue manager, see the following techdoc:

  - **Setup of a cluster and basic usage of clustered queues in WebSphere MQ 7**
  - Provides the details on how to setup a small WebSphere MQ cluster with 2 full repositories and 3 partial repositories located in 3 different hosts with different versions of MQ: 7.0, 7.1 and 7.5.
  - It also provides the details on the basic usage of clustered queues.
Setup of a small cluster - summary

- It is easier to setup a new cluster and to add members to the clusters by using the wizard provided by the MQ Explorer.

- If you want to use the runmqsc command, the following slides provide a summary.

- This WSTE will discuss all the steps in detail.
+++ Procedure for Setting up a new cluster:
1. Decide on the organization of the cluster and its name.
2. Decide which queue managers are to hold full repositories.
3. Alter the queue-manager definitions to add definitions for the full repositories.
   Example:
   ```sql
   ALTER QMGR REPOS('ClusterName')
   ```
4. Define the listeners that accept network requests from other queue managers.
   Example:
   ```sql
   DEFINE LISTENER(TCP.LISTENER) TRPTYPE(tcp) CONTROL(qmgr) PORT(1414)
   START LISTENER(TCP.LISTENER)
   ```
5. Define the CLUSRCVR channel for the queue managers, with a CONNAME pointing to itself.
   Generic Example:
   ```sql
   DEFINE CHANNEL(TO.channel_name) CHLTYPE(CLUSRCVR) +
   CONNAME('my_ip_name_or_address(port)') CLUSTER('ClusterName')
   ```
6. Define the CLUSSDR channel on the queue managers. The channel name must match that of the CLUSRCVR on the full repository.
   Generic Example:
   ```sql
   DEFINE CHANNEL(channel_name) CHLTYPE(CLUSSDR) +
   CONNAME('remote_ip_name_or_address(port)') CLUSTER('ClusterName')
   ```
+++ Summary of steps for adding a partial repository:

1. Define CLUSRCVR channel, with a CONNAME pointing to itself. This is the step that adds the partial repository into the cluster.
   - DEFINE CHANNEL(TO.channel_name) CHLTYPE(ClusRcvr) +
     CONNAME('my_ip_name_or_address(port)') CLUSTER('ClusterName')

2. Define a CLUSSDR channel for at least one full repository. The channel name must match that of the CLUSRCVR on the full repository.
   - DEFINE CHANNEL(channel_name) CHLTYPE(CluSSDr) +
     CONNAME('remote_ip_name_or_address(port)') CLUSTER('ClusterName')
Notes: Summary clustered queue and topic

+++ Summary of steps for adding a clustered queue

1. Define a clustered queue.
   Other queue managers in the cluster can send message to it without making remote-queue definitions for it.
   Only the local queue manager can read messages from an instance of the cluster queue.
   Generic Example:
   ```
   DEFINE QLOCAL(queueName) CLUSTER('ClusterName')
   ```

+++ Summary of steps for adding a clustered topic

1. Define a clustered topic
   Other queue managers in the cluster can publish and subscribe to it
   Generic Example:
   ```
   DEFINE TOPIC(TopicName) TOPICSTR('string') CLUSTER('ClusterName')
   ```
Using concrete examples in this WSTE

- It is useful to see concrete examples of the MQ commands.
  - Sometimes the generic format of the command may not answer all questions from customers.
- The commands used in this WSTE will use specific queue managers and queues.
- The names and versions of the queue managers are shown in the next page.
- Notice combination of platforms and versions.
  - Using Windows, Linux, and AIX
  - MQ 7.0, 7.1 and 7.5
Notes: Names of queue managers in test cluster

- Using Windows, Linux, and AIX, with MQ 7.0, 7.1 and 7.5
- It is a best practice to have the Full Repositories running at the highest (newest) version, in this case: **7.5.0.1**

- **QM_FR1** => Full Repository 1, Windows host, port 1441, MQ 7.5.0.1
  - Host-1 name: angelito.x.com

- **QM_PR1** => Partial Repository 1, Windows host, port 1442, MQ 7.0.1.10
  - Host-1 name: angelito.x.com
  - Note: This host exploits the multi-version feature introduced in 7.1.

- **QM_FR2** => Full Repository 2, Linux host, port 1443, MQ 7.5.0.1
  - Host-2 name: veracruz.x.com

- **QM_PR2** => Partial Repository 2, Linux host, port 1444, MQ 7.1.0.2
  - Host-2 name: veracruz.x.com
  - Note: This host exploits the multi-version feature introduced in 7.1.

- **QM_PR3** => Partial Repository 3, AIX host, port 1445, 7.5.0.0
  - Host-3 name: aemaix1.x.com
Topography - before adding qmgrs to cluster

- The topology before adding queue managers to the cluster looks like this:

**HOST-1: angelito**

- QM_FR1 (1441)
- QM_PR1 (1442)

**HOST-2: veracruz**

- QM_FR2 (1443)
- QM_PR2 (1444)

**HOST-3: aemaix1**

- QM_PR3 (1445)
Creating a cluster with 2 Full Repositories

- A full repository contains a complete set of information about every queue manager and object in the cluster.

- You will need at least one Full Repository.

- A best practice is to have two.

- This presentation shows a cluster with 2 Full Repositories.
Using MQ Explorer to add a full repository

- Let's use the MQ Explorer to create a cluster by specifying 2 Full Repositories:
  - QM_FR1 => Full Repository 1
  - QM_FR2 => Full Repository 2

- It is convenient if FIRST you add in the Navigation panel all the desired queue managers.
  - Click on "Queue Manager Clusters"
  - Then "New" and "Queue manager cluster …"
Using MQ Explorer to add a full repository

- You will see the dialog "Create Cluster".
- Enter the name of the cluster.
- In this example, we are using: "CLUSTER1"
Selecting first full repository

- Click Next to go to the step "Select the first full repository queue manager"
- Let's select: QM_FR1 and click Next.
Selecting second full repository

- Now let's select the "second full repository queue manager" by selecting: QM_FR2
Cluster-receiver channel for 1st FR

- Enter connection name (host-port) for the "cluster-receiver channel" for queue repository QM_FR1.
- The default name follows the template: TO.queuemanager (in this case TO.QM_FR1)
- In this example it is the name of the local host ANGELITO and port 1441.
Cluster-receiver channel for 2nd FR

- Do the same for the queue manager QM_FR2.
- The "cluster-receiver channel" is TO.QM_FR2
- In this example it is the name of the remote host veracruz.x.com and port 1443.
Where MQ Explorer shows the cluster

- You will see the new entry "CLUSTER1" under the "Queue Manager Clusters"
- Click the folder Full Repositories.
- Both repositories are listed on the right panel.
Initial topology of the cluster

- The topology of cluster CLUSTER1 looks like this.
- Note that the Cluster Sender (CLUSSDR) named (TO.QM_FR1) in QM_FR2 is created by the wizard and it matches the Cluster Receiver (CLUSRCVR) of QM_FR1 (TO.QM_FR1).
- Similarly for the other Cluster Sender channel
Notes: using line commands (1)

- Login as an MQ Administrator to the host where the queue managers are located

- 1) Alter the queue manager to indicate that a queue manager is going to be a full repository

  - Generic:
    - ALTER QMGR REPOS('ClusterName')

  - Note: This command is ONLY for the full repositories.

- Concrete examples:

  - runmqsc QM_FR1
    - ALTER QMGR REPOS('CLUSTER1')
    - end

  - runmqsc QM_FR2
    - ALTER QMGR REPOS('CLUSTER1')
    - end
Notes: using line commands (2)

- 2) Create cluster receiver (CLUSRCVR) channel objects

- This step is performed for all queue managers in the cluster.
- Every queue manager in the cluster needs a CLUSRCVR with a connection name (CONNAME) **pointing to itself**.

- Generic:
  - DEFINE CHANNEL(TO.channel_name) CHLTYPE(CLUSRCVR) +
  - CONNAME('my_ip_name_or_address(port)') CLUSTER('ClusterName')

- Concrete examples for the 2 Full Repositories.

  - runmqsc QM_FR1  # It resides in host angelito, CONNAME points to angelito
    - DEFINE CHANNEL(TO.QM_FR1) CHLTYPE(CLUSRCVR) +
    - CONNAME('angelito.x.com(1441)') CLUSTER('CLUSTER1')

  - runmqsc QM_FR2  # It resides in host veracruz, CONNAME points to veracruz
    - DEFINE CHANNEL(TO.QM_FR2) CHLTYPE(CLUSRCVR) +
    - CONNAME('veracruz.x.com(1443)') CLUSTER('CLUSTER1')
3) Create cluster sender (CLUSSDR) channel objects

Define one CLUSSDR to each full repository queue manager.

The channel name must match that of the cluster receiver (CLUSRCVR) on the other full repository.

**The CONNAME points to the OTHER repository.**

**Generic:**

```plaintext
DEFINE CHANNEL(channel_name) CHLTYPE(CLUSSDR) +
CONNAME('remote_ip_name_or_address(port)') CLUSTER('ClusterName')
```

**Concrete examples for the 2 Full Repositories.**

- runmqsc QM_FR1  # It resides in host angelito, CONNAME points to host veracruz
  ```plaintext
  DEFINE CHANNEL(TO.QM_FR2) CHLTYPE(CLUSSDR) +
  CONNAME('veracruz.x.com(1443)') CLUSTER('CLUSTER1')
  ```

- runmqsc QM_FR2  # It resides in host veracruz, CONNAME points to host angelito
  ```plaintext
  DEFINE CHANNEL(TO.QM_FR1) CHLTYPE(CLUSSDR) +
  CONNAME('angelito.x.com(1441)') CLUSTER('CLUSTER1')
  ```
Adding Partial Repositories

- Using MQ Explorer, from the “Queue Manager Clusters” section, right click and select:
  - Add Queue Manager To Cluster...
Select queue manager to add to cluster

- The wizard shows a list of queue managers that have not been added to the cluster.
- Select QM_PR1 and click Next.

- Specify that it is going to be a Partial Repository
Define cluster-receiver channel for PR

- Define the cluster-receiver channel: TO.QM_PR1
- Remember that the Connection Name points to itself! (Same queue manager name and port)
- QM_PR1 is located in host: angelito
- Thus, Connection Name is for angelito
Define cluster-sender channel for PR to FR

- Step 4 is to connect to a full repository. In this case it is QM_FR2 (because it is located in another host, to help reduce confusion with the commands)

- Step 5 is to define the cluster-sender channel: TO.QM_FR2

- Remember that the Connection Name points to a full repository
  - QM_PR1 is located in host: angelito
  - QM_FR2 is also located in host: veracruz
  - Thus, Connection Name is for host: veracruz
PRs in MQ Explorer

- Notice that QM_PR1 will be listed under the folder for "Partial Repositories"
The PR will be connected to both FRs

- After a new Partial Repository is connected to a Full Repository (QM_FR2)...

- ... this Full Repository (QM_FR2) will communicate the information on the new repository (QM_PR1) to the other Full Repository (QM_FR1).

- Auto-explicit cluster-sender channels will be created to connect QM_PR1 with QM_FR1.

- Thus, the Partial Repository will be eventually connected to both Full Repositories.
Topology so far (2 FRs, 1 PR)

- The topology so far looks like this:
Notes: using runmqsc to add PR

- Login as an MQ Administrator and issue:
  - $ runmqsc QM_PR2

- 1) Define a Cluster Receiver (CLUSRCVR) channel.
- Every queue manager in a cluster must define a cluster-receiver channel on which it can receive messages. The CONNAME points to itself!
- Note: This command adds the Partial repository to the cluster. There is no equivalent of “alter qmgr” for adding partial repositories.

  # QM_PR2 is located in host veracruz, thus CONNAME points to veracruz
  - DEFINE CHANNEL(TO.QM_PR2) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) +
    - CONNAME('veracruz.x.com(1444)') CLUSTER(CLUSTER1)

- 2) Define a CLUSSDR channel for at least one full repository (QM_FR1)
- Every queue manager in a cluster must define one cluster-sender channel to send messages to the full repositories.
- # QM_FR1 is located in host angelito, thus CONNAME points to angelito
  - DEFINE CHANNEL(TO.QM_FR1) CHLTYPE(CLUSSDR) TRPTYPE(TCP) +
    - CONNAME('angelito.x.com(1441)') CLUSTER(CLUSTER1)
Topology after adding the 3 PRs

- Continue adding the remaining partial repositories
- Notice that at this point there are NO connections between PRs. (they will be added later as needed)
- The topology now looks like this:
More cluster info from the MQ Explorer

- Let's explore some aspects of a Partial Repository:
- Click on the tab "Cluster-sender Channels" to see the 2 channels that connect this partial repository to the full repositories.
- The other tabs are for cluster queues, cluster topics and Cluster-receiver Channels.
Notes: using runmqsc to verify cluster info

- If there is a value for REPOS attribute for the queue manager, then it indicates that the queue manager is a Full Repository for the specified cluster.

- runmqsc QM_FR1
- display qmgr repos
- AMQ8408: Display Queue Manager details.
  - QMNAME(QM_FR1) \(\text{REPOS}^{\text{CLUSTER1}}\)

- The following command provides more details from the perspective of a cluster.
  - display clusqmgr(QM_FR1)

- At this point, the relevant attributes are shown below:
  - CLUSQMGR(QM_FR1) \(\text{CLUSTER}^{\text{CLUSTER1}}\)
  - CHANNEL(TO.QM_FR1) \(\text{CONNAME}^{\text{ANGELITO(1441)}}\)
  - DEFTYPE(CLUSRCVR) \(\text{QMTYPE}^{\text{REPOS}}\) => Full Repository
  - QMID(QM_FR1_2013-03-22_12.48.28) \(\Rightarrow\) Unique identifier for the queue manager to distinguish this queue manager instance from other queue managers that might have previously existed in a queue manager cluster with the same name.
Notes: using runmqsc to verify cluster info

- Display the cluster sender and cluster receiver channels, showing the cluster name and the connection name.

- display channel(*) CHLTYPE(CLUSSDR) cluster conname
  - AMQ8414: Display Channel details.
  - CHANNEL(TO.QM_FR2) CHLTYPE(CLUSSDR)
  - CLUSTER(CLUSTER1)
  - CONNAME(veracruz.x.com(1443))

- display channel(*) CHLTYPE(CLUSRCVR) cluster conname
  - AMQ8414: Display Channel details.
  - CHANNEL(TO.QM_FR1) CHLTYPE(CLUSRCVR)
  - CLUSTER(CLUSTER1)
  - CONNAME(ANGELITO(1441))
Let's explore specific details for a Partial Repository
runmqsc QM_PR1

Notice that a null value in REPOS indicates that this is NOT a Full Repository and consequently, it is a Partial Repository.

display qmgr repos
QMNAME(QM_PR1) REPOS()

To find out the cluster to which this queue manager belongs to, it is necessary to use the following command. Notice the attribute-value pair: CLUSTER(CLUSTER1)
The QMTYPE attribute has a value of NORMAL, which indicates that this is NOT a Full Repository and consequently, it is a Partial Repository.
A Full Repository will have as value "REPOS".

display clusqmgr(QM_PR1)
CLUSQMGR(QM_PR1) CHANNEL(TO.QM_PR1)
CLUSTER(CLUSTER1) CLUSTIME(13.03.01)
CONNAME(angelito.x.com(1442))
QMID(QM_PR1_2013-03-22_12.51.5 QMTYPE(NORMAL))
Local queue, not yet a clustered queue

- It is important to understand that **local queues** defined in one queue manager are NOT made visible to the other members of the cluster.
- Just because a queue is defined in a queue manager in the cluster, does not make the queue a clustered queue.

- The local queue CQ1 is defined in the queue manager QM_PR1, but it has NOT being designated to be a clustered queue.
  - runmqsc QM_PR1
  - DEFINE QLOCAL(CQ1)
Expected behavior of Puts and Gets

- **OK.** application (such as amqsputc) can put a message to the local CQ1 when connected to QM_PR1.

- **OK.** application (such as amqsgetc) can get a message from the local CQ1 when connected to QM_PR1.

- **FAILED.** application CANNOT put a message into CQ1 when connected to another queue manager, such as QM_PR2.

- **FAILED.** application CANNOT get a message from CQ1 when connected to another queue manager, such as QM_PR2.
Expected behavior of non-clustered queue

HOST-1: angelito

Full: QM_FR1

Partial: QM_PR1

Local Q: CQ1

OK

PUT

GET

OK

PUT

GET

HOST-2: veracruz

Full: QM_FR2

Partial: QM_PR2

Failed

PUT

GET

Failed
Notes: testing of local queue (not clustered)

- Login as a user that has proper authority to connect to the queue manager and to put messages. For simplicity in this test scenario, you can login as an MQ Administrator.

- Window 1: (Put in Windows - QM_PR1)
  - Use the amqsputc sample to put a message into CQ1 in QM_PR1
  - C:\> set MQSERVER=SYSTEM.ADMIN.SVRCONN/TCP/angelito(1442)
  - C:\> amqsputc CQ1 QM_PR1
  - Sample AMQSPUT0 start
  - target queue is CQ1
  - TEST-PUT-1 (then press ENTER)
  - (do not type anything, then press ENTER to exit)
  - Sample AMQSPUT0 end

- Window 2: (Get in Windows - QM_PR1)
  - Use the amqsgetc sample to get a message from CQ1 in QM_PR1
  - C:\> set MQSERVER=SYSTEM.ADMIN.SVRCONN/TCP/angelito(1442)
  - C:\> amqsgetc CQ1 QM_PR1
  - Sample AMQSGET0 start
  - message <TEST-PUT-1>
  - no more messages
  - Sample AMQSGET0 end
Notes: testing of local queue (not clustered)

- Unsuccessful Put from local queue in QM_PR1 (not yet part of the cluster) when using another queue manager from the cluster

- Window 3: (Put in Linux - QM_PR2)
  - Notice that the queue CQ1 is NOT defined in QM_PR2.

- `$ export MQSERVER=SYSTEM.ADMIN.SVRCONN/TCP/'veracruz.x.com(1444)'

- `$ amqsputc CQ1 QM_PR2
  - Sample AMQSPUT0 start
  - target queue is CQ1
  - MQOPEN ended with reason code 2085
  - unable to open queue for output
  - Sample AMQSPUT0 end

- The return code 2085 indicates that the queue manager QM_PR2 does not have a queue CQ1.

- `$ mqrc 2085
  - 2085 0x00000825 MQRC_UNKNOWN_OBJECT_NAME`
Local queue added to the cluster

- In order to make a queue visible to the other members of the cluster, it is necessary to ALTER the queue and specify the cluster name:
  - `ALTER QLOCAL(CQ1) CLUSTER('CLUSTER1')`

- You do a remote PUT into a clustered queue.
- However, you cannot do a remote GET from a clustered queue.
- The GET can ONLY be done from the local queue.
The queue CQ1 is a LOCAL queue in QM_PR1.
Notice that this queue CQ1 now belongs to CLUSTER(CLUSTER1).
runmqsc QM_PR1
DISPLAY QLOCAL(CQ1) CLUSTER
  QUEUE(CQ1)  TYPE(QLOCAL)
  CLUSTER(CLUSTER1)

The queue CQ1 is NOT a local queue in QM_PR2. It is a clustered queue.
runmqsc QM_PR2
DISPLAY QLOCAL(CQ1) CLUSTER
AMQ8147: WebSphere MQ object CQ1 not found.

You need to run the specific command for clustered queues.
Notice that the queue is LOCAL to QM_PR1 (attributes CLUSQT, CLUSTQMGR)

DISPLAY QCLUSTER(CQ1)
  QUEUE(CQ1)  TYPE(QCLUSTER)
  CLUSDATE(2013-05-08)  CLUSTER(CLUSTER1)
  CLUSQMGR(QM_PR1)  CLUSQT(QLOCAL)
  QMID(QM_PR1_2013-05-06_15.14.28)
Expected behavior of Puts and Gets

- **OK.** application (such as amqspuct) can put a message to the local CQ1 when connected to QM_PR1.
- **OK.** application (such as amqsgetc) can get a message from the local CQ1 when connected to QM_PR1.
- **OK.** application CAN put a message into CQ1 when connected to another queue manager, such as QM_PR2.
- **FAILED.** application CANNOT get a message from CQ1 when connected to another queue manager, such as QM_PR2.
Expected behavior for clustered queues

**HOST-1: angelito**
- Full: QM_FR1
  - Local Clustered Q: CQ1
- Partial: QM_PR1
  - PUT
  - GET

**HOST-2: veracruz**
- Full: QM_FR2
  - Clusters Q: CQ1
- Partial: QM_PR2
  - PUT
  - GET
Notes: testing of local queue (not clustered)

- Window 3: (Put in Linux - QM_PR2)
- Notice that the queue CQ1 is NOT defined in QM_PR2.

- $ export MQSERVER=SYSTEM.ADMIN.SVRCONN/TCP/'veracruz.x.com(1444)'
- $ amqsputc CQ1 QM_PR2
- Sample AMQSPUT0 start
- target queue is CQ1
- TEST-PUT-3
- Sample AMQSPUT0 end

- Notice that the message with text "TEST-PUT-3" was successfully placed in the queue CQ1 in the remote partial repository QM_PR2.
- But this queue manager does NOT have a local queue named CQ1!
- Instead, this queue manager has a "cluster queue" CQ1 that points to the queue manager (QM_PR1) of where a local queue is located and which can store messages.
- The clustering code will create an auto-defined sender channel from QM_PR2 to QM_PR1 called “TO.QM_PR1: and send the message to the queue CQ1 in QM_PR1.
Because the new channel “TO.QM_PR1” was created automatically in QM_PR2, you will NOT be able to see the information for this channel using the following command:

- runmqsc QM_PR2
- DISPLAY CHANNEL(TO.QM_PR1)
- AMQ8147: WebSphere MQ object TO.QM_PR1 not found.

Instead, you have to use the following command:

- DISPLAY CHSTATUS(TO.QM_PR1)
  - CHANNEL(TO.QM_PR1)                      CHLTYPE(CLUSSDR)
  - CONNAME(angelito.x.com(1442))
  - CURRENT                                 RQMNAME(QM_PR1)
  - STATUS(RUNNING)                        SUBSTATE( )
  - XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
Topography after adding auto channel

- After the auto channel TO.QM_PR1 is added in QM_PR2 to connect to QM_PR1, the topology now looks like this:
SYSTEM.CLUSTER.TRANSMIT.QUEUE

- When the message is put into the definition of the clustered queue CQ1 in QM_PR2, the message is forwarded to a special clustering queue called: SYSTEM.CLUSTER.TRANSMIT.QUEUE
- Then the message travels via the newly created channel TO.QM_PR1 to QM_PR1 which has the local queue.
Topology after adding auto channel

- The auto defined channel was added “on-demand” between QM_PR2 and QM_PR1
- At this point, there are NO auto defined channels regarding QM_PR3.
- Also, QM_PR3 does not know yet about CQ1.
Authority to Put message in clustered queue

- A non-MQ administrator will get an error when trying to put a message into a clustered queue lack of proper authority: reason code 2035 (MQRC_NOT_AUTHORIZED)
There are 3 solutions for allowing a non-MQ administrator to put a message into a clustered queue.

- The local queue CQ1 resides in QM_PR1.
- The definition of the clustered queue CQ1 is in QM_PR2.
- The non-MQ administrator will do a put into CQ1 in QM_PR2.

Need to give connection access in QM_PR2 to the non-MQ Administrator

```
setmqaut -m QM_PR2 -t qmgr -g mqusers +connect +inq +dsp
```

- a) Give local authority for SYSTEM.CLUSTER.TRANSMIT.QUEUE in QM_PR2
  ```
  setmqaut -m QM_PR2 -t queue -n SYSTEM.CLUSTER.TRANSMIT.QUEUE -g mqusers +put
  ```

- b) Give using local authority for clustered queue CQ1 in QM_PR2
  ```
  setmqaut -m QM_PR2 -t queue -n CQ1 -g mqusers +put
  ```

- c) Use ClusterQueueAccessControl=RQMName in qm.ini (new in 7.1)

For more information see:

- WebSphere MQ 7.5 > Configuring > Configuring a queue manager cluster
- Access control and multiple cluster transmission queues
Topics in a Cluster

- For the handling of Topics in a cluster see the following techdoc:
  - **Basic usage of clustered topics in WebSphere MQ 7**
  - The focus is on basic usage of clustered topics in WebSphere MQ 7.x.
  - First, the scenario of a non-clustered topic is explored regarding publishing and subscribing.
  - Then the topic is included in a cluster.
  - It discusses durable and non-durable subscriptions
Creating a topic (non clustered)

- Using MQ Explorer, create 1 topic.
- Topic name "T1" which has a topic string "sports"
- Using runmqsc:
  - define topic(T1) topicstr('sports')
- For now, it is not incorporated into the cluster.
- MQ Explorer: right click on Topics, then New then Topic
Expected behavior of non-clustered topics

- **OK.** application such as "amqspubc" can publish a message to the local topic T1 in QM_PR1.

- **OK.** A subscriber application such as "amqssubc" can get a message from local topic T in QM_PR1.

- **FAILED.** Another application CAN publish a message into T1 when connected to another queue manager, such as QM_PR2, but it is NOT really the same topic object!

- **FAILED.** Another subscriber application CANNOT get a message from T1 from QM_PR1 when connected to another queue manager, such as QM_PR2.
Expected behavior of non-clustered topics

HOST-1: angelito
Full: QM_FR1
Partial: QM_PR1
T1 - sports

OK
PUB
SUB

OK

HOST-2: veracruz
Full: QM_FR2
Partial: QM_PR2

FAILED
PUB
SUB

FAILED
Open 2 command prompts.
We need to start first the subscriber.
For simplicity, this subscriber is “non-durable”.

set MQSERVER=SYSTEM.ADMIN.SVRCOONE/TCP/angelito(1442)

Window 1: C:\> amqssubc sports QM_PR1
Sample AMQSSUBA start
Calling MQGET : 30 seconds wait time

Window 2: C:\ amqspubc sports QM_PR1
Sample AMQSPUBA start
target topic is sports
test-1 (press ENTER)
(press ENTER to end)
Sample AMQSPUBA end

Back to Window 1 to see received message:
message <test-1>
Calling MQGET : 30 seconds wait time
Notes: caution when using local vs cluster T

- Notice that because the topic T1 is NOT yet a clustered topic, then when a publisher tries to publish to the topic T1 (string 'sports') in QM_PR2, then the publishing will be successful, but this is not the same topic that is going to be clustered.

  - WebSphere MQ > Planning > Designing a WebSphere MQ architecture > Introduction to publish/subscribe > Distributed publish/subscribe > Publish/subscribe topologies > Publish/Subscribe Clusters > Cluster topics
  - Multiple cluster topic definitions

- A local topic definition overrides a remotely defined cluster topic definition of the same name. Creation of multiple definitions of the same cluster topic on different queue managers in a cluster is also possible. Both of these scenarios require some caution however, the reasons are explained in this topic.

- Just as for clustered queues, having multiple definitions of the same cluster topic object in a cluster introduces the possibility of different properties defined on each. It is not easy to determine which version of the topic definition is seen by each queue manager in the cluster and it is therefore hard to determine the expected behavior.
Notes: authorities for topics

- Technote: Authorizations needed for non-mqm users to publish and subscribe to Topics in MQ V7

- Summary

- If you want to let the users in the groups "editors" and "journalists" connect to the queue manager WMQ7:
  - setmqaut -m QMGR -t qmgr -g editors +connect +inq +dsp
  - setmqaut -m QMGR -t qmgr -g journalists +connect +inq +dsp

- If you want to let the users in the group "editors" subscribe to topic “T1” on Queue Manager QMGR and to resume durable subscriptions:
  - setmqaut -m QMGR -n T1 -t topic -g editors +sub +resume

- If you want to allow users in the group "journalists" to publish to the topic:
  - setmqaut -m QMGR -n T1 -t topic -g journalists +pub
Adding topic to the cluster

- Using MQ Explorer, alter the topic T1 to specify that it is part of the cluster.
- Using runmqsc:
  - ALTER TOPIC(T1) CLUSTER(CLUSTER1)
The clustered topics are fanned out to all

- In contrast to clustered queues (only when a PR needs to use it, gets a copy of the definition),
- A clustered topic is fanned out to ALL the queue managers in a cluster.
Expected behavior of clustered topics

- **OK.** Application such as "amqspubc" can publish a message to the local topic T1 in QM_PR1.

- **OK.** A subscriber application such as "amqssubc" can get a message from local topic T in QM_PR1.

- **OK.** Another application can publish a message into T1 when connected to another queue manager, such as QM_PR2.

- **OK.** Another subscriber application can get a message from T1 when connected to another queue manager, such as QM_PR2.
Expected behavior of clustered topics

HOST-1: angelito

- Full: QM_FR1
  - Clustered T1 string: sports
- Partial: QM_PR1
  - Local Clustered T1 string: sports
  - OK
  - PUB
  - OK
  - SUB

HOST-2: veracruz

- Full: QM_FR2
  - Clustered T1 string: sports
- Partial: QM_PR2
  - OK
  - PUB
  - OK
  - SUB
The clustered topic T1 was defined in QM_PR1.

In QM_PR1 you can issue the following runmqsc command to get more info:
- DISPLAY TOPIC(T1)

However, if you use another queue manager, such as QM_PR2, the above command will fail:
- DISPLAY TOPIC(T1)
- AMQ8147: WebSphere MQ object T1 not found.

In QM_PR2 you will need to issue:
- DISPLAY TCLUSTER(T1)
- AMQ8633: Display topic details.
- TOPIC(T1) TYPE(CLUSTER)
- TOPICSTR(sports) DESCR( )
- CLUSTER(CLUSTER1) CLUSQMGR(QM_PR1)
- QMID(QM_PR1_2013-05-06_15.14.28) DURSUB(ASPARENT)
- ... NPMSGDLV(ASPARENT) PUBSCOPE(ASPARENT)
- SUBSCOPE(ASPARENT) PROXYSUB(FIRSTUSE)
- WILDCARD(PASSTHRU) ...
Additional WebSphere Product Resources

- Learn about upcoming WebSphere Support Technical Exchange webcasts, and access previously recorded presentations at: http://www.ibm.com/software/websphere/support/supp_tech.html

- Discover the latest trends in WebSphere Technology and implementation, participate in technically-focused briefings, webcasts and podcasts at: http://www.ibm.com/developerworks/websphere/community/

- Join the Global WebSphere Community: http://www.websphereusergroup.org

- Access key product show-me demos and tutorials by visiting IBM® Education Assistant: http://www.ibm.com/software/info/education/assistant


- Sign up to receive weekly technical My Notifications emails: http://www.ibm.com/software/support/einfo.html
Connect with us!

1. **Get notified on upcoming webcasts**
   Send an e-mail to wsehelp@us.ibm.com with subject line “wste subscribe” to get a list of mailing lists and to subscribe

2. **Tell us what you want to learn**
   Send us suggestions for future topics or improvements about our webcasts to wsehelp@us.ibm.com

3. **Be connected!**
   Connect with us on Facebook
   Connect with us on Twitter
Questions and Answers