
CREATING MQ CLUSTER

TABLE OF CONTENTS

Definitions of different terms.....	2
Cluster.....	2
Cluster queue manager	2
Cluster queue	3
Repository.....	3
Comparison with distributed queuing.....	3
Setting up a new cluster	5
Steps Summary.....	6
Detail of steps.....	7
Decide on the organization of the cluster and its name.....	7
Determine which queue managers should hold full repositories.....	7
Alter the queue-manager definitions to add repository definitions.....	7
Define the CLUSRCVR channels	8
Define the CLUSSDR channels.....	9
Define the cluster queue INVENTQ	9
Start the listener on both Queue Managers.....	10
Verify and test the cluster.....	10
Using the sample program:.....	10
References	11

Definitions of different terms

Prior to going into details of cluster making, we will discuss brief definitions of different terms which need to be known and clarified before creating a cluster.

Cluster

A cluster is a network of queue managers that are logically associated in some way. The queue managers in a cluster may be physically remote. For example, they might represent the branches of an international chain store and be physically located in different countries. Each cluster within an enterprise should have a unique name

Cluster queue manager

A cluster queue manager is a queue manager that is a member of a cluster. A queue manager may be a member of more than one cluster. Each cluster queue manager must have a name that is unique throughout all the clusters of which it is a member. A cluster queue manager can host queues, which it *advertises* to the other queue managers in the cluster. A cluster queue manager does not have to host or advertise any queues. It can just feed messages into the cluster and receive only responses that are directed explicitly to it, and not to advertised queues.

Cluster queue

A cluster queue is a queue that is hosted by a cluster queue manager and made available to other queue managers in the cluster. The cluster queue manager makes a local queue definition for the queue, specifying the name of the cluster where the queue is to be found. This definition has the effect of showing the other queue managers in the cluster that the queue is there. The other queue managers in the cluster can put messages to a cluster queue without needing a corresponding remote-queue definition. A cluster queue can be advertised in more than one cluster.

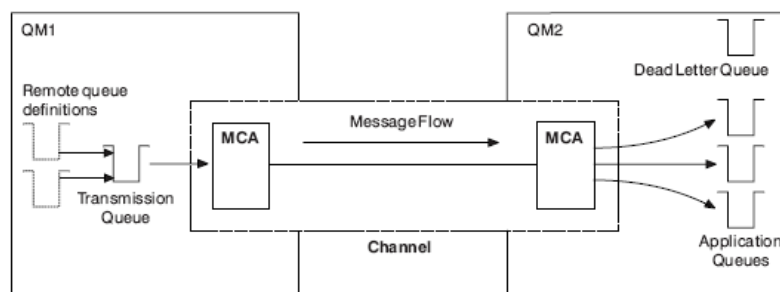
Repository

A repository is a collection of information about the queue managers that are members of a cluster. This information includes queue manager names, their locations, their channels, which queues they host, and so on. The information is stored in the form of messages on a queue called `SYSTEM.CLUSTER.REPOSITORY.QUEUE`. (This queue is one of the default objects created when you start a WebSphere MQ queue manager, except on WebSphere MQ for z/OS where it is defined as part of queue manager customization.) Typically, two queue managers in a cluster hold a *full repository*.

Comparison with distributed queuing

If you do not use clusters, your queue managers are independent and communicate using distributed queuing. If one queue manager needs to send messages to another it must have defined:

- A transmission queue
- A channel to the remote queue manager



CREATING MQ CLUSTER

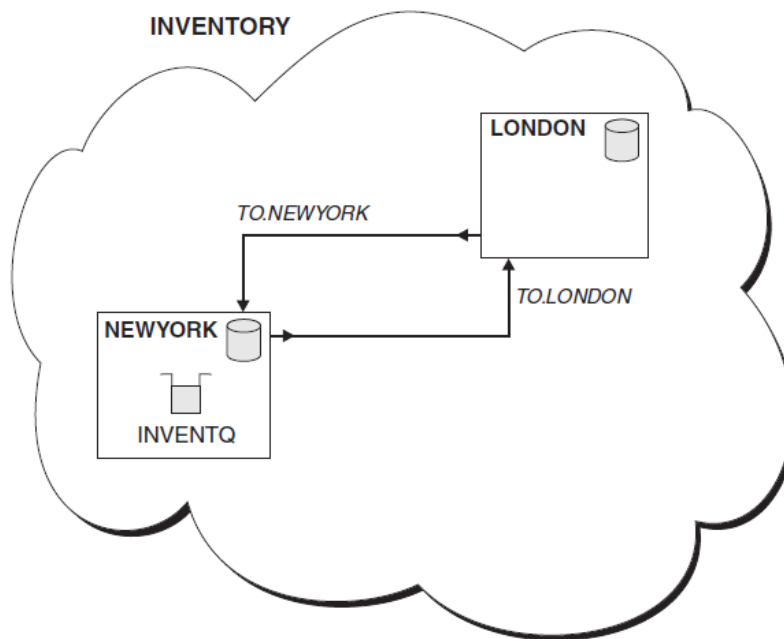
Microsoft Windows

If you group queue managers in a cluster, the queue managers can make the queues that they host available to every other queue manager in the cluster. Any queue manager can send a message to any other queue manager in the same cluster without explicit channel definitions, remote-queue definitions, or transmission queues for each destination. Every queue manager in a cluster has a single transmission queue from which it can transmit messages to any other queue manager in the cluster. Each queue manager in a cluster needs to define only:

- One cluster-receiver channel on which to receive messages
- One cluster-sender channel with which it introduces itself and learns about the cluster

Setting up a new cluster

Suppose you are setting up a new WebSphere MQ network for a chain store. The store has two branches, one in London and one in New York. The data and applications for each store are hosted by systems running separate queue managers. The two queue managers are called LONDON and NEWYORK. The inventory application runs on the system in New York, connected to queue manager NEWYORK. The application is driven by the arrival of messages on the INVENTQ queue, hosted by NEWYORK. The two queue managers, LONDON and NEWYORK, are to be linked in a cluster called INVENTORY so that they can both put messages to the INVENTQ.



Examples shows what this cluster looks like.

Note: On WebSphere MQ for Windows you can use one of the wizards supplied with WebSphere MQ Explorer to create a new cluster similar to the one created by this task.

Steps Summary

Following are the summary of the steps you have to follow to successfully create a cluster.

1. Decide on the organization of the cluster and its name
2. Determine which queue managers should hold full repositories
3. Alter the queue-manager definitions to add repository definitions
4. Define the CLUSRCVR channels
5. Define the CLUSSDR channels
6. Define the cluster queue INVENTQ
7. Starting listeners
8. Verify and test the cluster

Detail of steps

Decide on the organization of the cluster and its name

You have decided to link the two queue managers, LONDON and NEWYORK, into a cluster. A cluster with only two queue managers offers only marginal benefit over a network that is to use distributed queuing, but that is extendable. When you open new branches of your store, you will be able to add the new queue managers to the cluster easily and without any disruption to the existing network. Here, the cluster name we will use is INVENTORY.

Determine which queue managers should hold full repositories

In any cluster, you need to nominate at least one queue manager, or preferably two, to hold full repositories. In this example there are only two queue managers, LONDON and NEWYORK, both of which hold full repositories.

Alter the queue-manager definitions to add repository definitions

On each queue manager that is to hold a full repository, you need to alter the queue-manager definition, using the ALTER QMGR command and specifying the REPOS attribute:

```
C:\> runmqsc LONDON
ALTER QMGR REPOS(INVENTORY)
AMQ8005: Websphere MQ queue manager changed.

C:\> runmqsc NEWYORK
ALTER QMGR REPOS(INVENTORY)
AMQ8005: Websphere MQ queue manager changed.
```

Queue managers LONDON and NEWYORK will be changed to a full repository.

Note: If you just runmqsc and enter the ALTER QMGR command, the default queue manager will be changed.

Define the CLUSRCVR channels

On every queue manager in a cluster you need to define a cluster-receiver channel on which the queue manager can receive messages.

This definition defines the queue manager's connection name and the CLUSTER keyword shows the queue manager's availability to receive messages from other queue managers in the cluster. The queue manager's connection name is stored in the repositories, where other queue managers can refer to it. Examples of the responses to the commands are shown below.

Setting up a cluster (Using transport protocol TCP/IP)

On the LONDON queue manager, define:

```
DEFINE CHANNEL(TO.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-receiver channel for queue manager LONDON')
```

In this example the channel name is TO.LONDON, and the connection name (CONNNAME) is the network address of the machine the queue manager resides on, which is LONDON.CHSTORE.COM.

(The network address can be entered either as an alphanumeric DNS hostname, or as a dotted-decimal IP address.)

```
Eg. CONNNAME(10.1.234.112(1414))
```

Where 1414 is the port on which queue manager is listening.

Do not allow the CONNNAME to specify a generic name.

On the NEWYORK queue manager, define:

```
DEFINE CHANNEL(TO.NEWYORK) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-receiver channel for queue manager NEWYORK')
```


Define the CLUSSDR channels

On every queue manager in a cluster you need to define one cluster-sender channel on which the queue manager can send messages to one of the full repository queue managers. In this case there are only two queue managers, both of which hold full repositories. They must each have a CLUSSDR definition that points to the CLUSRCVR channel defined at the other queue manager.

Note that the channel names given on the CLUSSDR definitions must match those on the corresponding CLUSRCVR definitions.

On the LONDON queue manager, define:

```
DEFINE CHANNEL(TO.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-sender channel from LONDON to repository at NEWYORK')
```

On the NEWYORK queue manager, define:

```
DEFINE CHANNEL(TO.LONDON) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-sender channel from NEWYORK to repository at LONDON')
```

Once a queue manager has definitions for both a cluster-receiver channel and a cluster-sender channel in the same cluster, the cluster-sender channel is started.

Define the cluster queue INVENTQ

Define the INVENTQ queue on the NEWYORK queue manager, specifying the CLUSTER keyword.

```
DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
```

The CLUSTER keyword causes the queue to be advertised to the cluster. As soon as the queue is defined it becomes available to the other queue managers in the cluster. They can send messages to it without having to make a remote-queue definition for it.

Start the listener on both Queue Managers

Now that you have completed all the definitions, start a listener program on each queue manager. The listener program listens for incoming network requests and starts the cluster-receiver channel when it is needed.

The cluster set up by this task looks like this:

```
DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
AMQ8006: Websphere MQ queue created.
```

Clearly, this is a very small cluster. However, it is useful as a proof of concept. The important thing to understand about this cluster is the scope it offers for future enhancement.

Verify and test the cluster

Issue some DISPLAY commands to verify the cluster that you have set up. The responses you see should be similar to those shown below.

From the NEWYORK queue manager, issue the command:

```
dis clusqmgr(*)
```

Now issue the corresponding DISPLAY CHANNEL STATUS command:

```
dis chstatus(*)
```

Because the INVENTQ queue has been advertised to the cluster, applications running on NEWYORK and applications running on LONDON both can put messages to the INVENTQ queue. They can receive responses to their messages by providing a reply-to queue and specifying its name when they put messages.

Using the sample program:

Test your setup by sending some messages between the two queue managers, using amqspout. In the following example LONDON puts a message to the INVENTQ at NEWYORK:

1. On LONDON issue the command:

```
amqsput INVENTQ LONDON
```

2. Type some messages

3. On NEW YORK issue the command:

```
amqsget INVENTQ NEWYORK
```

4. You should now see the messages you entered on LONDON

References

- IBM Document
Websphere MQ Queue Manager Clusters
SC34-6589-00
URL : <http://www-01.ibm.com/support/docview.wss?uid=pub1sc34658900>



© Copyright IBM Corporation 2010
IBM Global Services
Route 100
Somers, NY 10589
U.S.A.
Produced in the United States of America
08-10
All Rights Reserved

IBM, the IBM logo, ibm.com, Lotus®, Rational®, Tivoli®, DB2® and WebSphere® are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at ibm.com/legal/copytrade.shtml. Other company, product and service names may be trademarks or service marks of others. The information contained in this documentation is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this documentation, 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 documentation or any other documentation. Nothing contained in this documentation 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 the applicable license agreement governing the use of IBM software. This document illustrates how one organization uses IBM products. Many factors have contributed to the results and benefits described; IBM does not guarantee comparable results elsewhere.