MQSeries Clustering on Windows:

Defining an MQSeries Cluster

Revision:
1.0

Prepared by:
Scott Meridew – Director – Professional Services,
TxMQ Inc.

www.txmq.com

scott@txmq.com
888-519-7791 x334
# TABLE OF CONTENTS

1 CONVENTIONS .................................................................................................................................3

2 AUDIENCE ..........................................................................................................................................3

3 CREATING AN MQSERIES CLUSTER ..................................................................................................4

4 DEFINING THE CLUSTER USING MQSERIES EXPLORER ..................................................................5

   4.1 CREATING THE QUEUE MANAGER ..........................................................................................5

   4.2 CREATING THE CLUSTER .........................................................................................................10

   4.3 CREATING AND TESTING A DISTRIBUTED CLUSTER QUEUE ...................................................16

   4.4 JOINING AN EXISTING CLUSTER ............................................................................................21
MQSeries Clustering on Windows

1 Conventions

Throughout this document formatting has been used to indicate the following to the reader:

- Important notes are in **bold italics**
- Definitions and specific new terms are in *italics*
- MQSeries MQSC commands, objects, and API calls are in **BOLD UPPERCASE**.
- URLs, FTP and email addresses are in **blue** when displayed from a monitor. For print, http:// always proceeds a URL, ftp:// always proceeds an FTP address, and mailto: always proceeds an email address.

2 Audience

This document will be useful for both MQSeries developers who are implementing an MQSeries cluster, as well as system architects and designers.
MQSeries Clustering on Windows

3 Creating an MQSeries cluster

With the introduction of MQSeries for WindowsNT V5.1, IBM has included graphical utilities for the management of MQSeries. However, since MQSeries is a multi-platform application, much of the IBM documentation on clustering shows how to create and manage clusters using the command-line interface (runmqsc). In fact, the ‘Queue Manager Clusters’ document in the MQSeries for WindowsNT V5.1 library does not have any GUI screen shots showing how to build or manage an MQSeries cluster. This document will guide you through the configuration and management of an MQSeries cluster using the graphical tools provided in MQSeries for NT V5.1.

The graphical tools provided with MQSeries Explorer for WindowsNT V5.1 include:

- **MQSeries Explorer** – a Microsoft Management Console snap-in that allows an administrator to add/remove/manage MQSeries objects (both local and remote).
- **MQSeries Services** – a Microsoft Management Console snap-in that allows an administrator to add/remote/manage MQSeries services such as listeners, traces, trigger monitors, and channel initiators.
- **MQSeries First Steps** – A utility that enables you to define a default configuration, test an existing configuration using a simple ‘postcard’ application, or test specific objects using an API exerciser.

There are essentially three ways to create an MQSeries cluster on WindowsNT; the hard way, the easy way, and the quick way.

**The Hard Way**

The hard way to create a cluster on NT is to create all the objects using the command line interface (runmqsc), which is still available in V5.1. Having said that, you may be working in a mixed-mode environment where you maintain mqsc scripts centrally for all your distributed systems. In this case, you’ll have to update and test your scripts anyway, so you might opt for this approach.

**The Quick Way**

As usual, the quick way is not necessarily the best way. Using the MQSeries first steps default configuration wizard, you can allow MQSeries to define your MQSeries objects, as it poses as series of questions about your environment. The trouble with using the default configuration is that the MQSeries object names are defined automatically by the utility, and likely will not conform to your internal naming conventions. This tool is useful to get you going quickly and to test the clustering functionality, but ultimately, you’ll want to do it the ‘Easy Way’.

**The Easy Way**

The easy way is to use the MQSeries Explorer tool to create the cluster, the queue managers, and the cluster channels. If you are already familiar with the Microsoft Management Console (MMC), this process will be even easier. This document will focus on this approach and provides step-by-step instructions on creating a 3-node WindowsNT cluster.
4 Defining the cluster using MQSeries Explorer

In this section, we’ll create a three-node Windows-NT cluster from scratch. The machines used are all Pentium-II based systems running WindowsNT 4.0 SP5. Although you may already have queue managers defined, these steps will show you how to create a queue manager first. The three machines used in this sample are cgilab72 (repository queue manager #1), cgilab71 (repository queue manager #2), and finally cgilab221 (a non-repository queue manager). The machines are all members of the same NT domain (CGILAB) and are all logged on using a domain account called mqmadmin, which is a member of the mqm domain group.

The first step is to decide on the names of your MQSeries objects, preferably according to some kind of standard naming convention. Defining connection names is where you will have to make an important decision, based on your network configuration. We could have used the IP address of cgilab71, the NetBIOS name (since our network uses WINS for NetBIOS resolution). However, since we are also using DNS for hostname resolution, we defined the connections names as fully qualified domain names (i.e. cgilab71.on.bell.ca).

The naming/conventions used in these examples are as follows:

Queue Managers: QM_<hostname>
Channels: TO_<hostname>
Cluster: CGILAB
Connection Names: Fully Qualified Domain name (FQDN) - for example: cgilab71.on.bell.ca
Cluster Queue for testing: clq_test

4.1 Creating the queue manager

The first step, if you don’t already have one, is to create a queue manager using the MQSeries Explorer tool. Open the program up by selecting Start…Programs…IBM MQSeries…MQSeries Explorer. Then right click on Queue Managers, and select New as show in Figure 1.
MQSeries Clustering on Windows

Figure 1
The Create Queue Manager wizard starts up and prompts for the queue manager attributes. Enter the queue manager name as shown in figure 2, and select ‘make this the default queue manager’. Click next. Click next again on the following dialog box (Step 2) to accept the default logging values.

**Figure 2**

Be sure to select the ‘Create Server Connection Channel to allow remote administration of the queue manager over TCP/IP’ in Step 3. We were unable to create the cluster without selecting this option. This is also needed if you want to remotely manage or view cluster queue managers from within the MQSeries Explorer window. Figure 3 shows this dialog box.
MQSeries Clustering on Windows

Figure 3

The final step in creating the queue manager is to enable the channel listener and specify the TCP/IP port that MQSeries will be using. We selected the default port of 1414 as shown in Figure 4. Click ‘Finish’ to create the queue manager.

Figure 4

The queue manager needs a listener to monitor for incoming network connections, for some network protocols.

Create Queue Manager (Step 4)

- Queue Manager: QM_cglist72

The listener needs to listen on a port number not used by any other queue manager, service or application on this computer.

Listen on port number: 1414
Important Note: These steps must be performed for at least two systems, before you can create the cluster. As you will see, when the cluster is created, it will request information about the ‘second’ repository queue manager. We created the second repository queue manager as QM_cgilab71 on node cgilab71 in our lab.
4.2 Creating the cluster

The next step is to create the cluster. In MQSeries Explorer, right click the Cluster folder and select New…Cluster as shown in figure 5.

Figure 5
This will launch the ‘create cluster’ wizard, which will prompt you for the cluster name, channel names, repository queue manager names, and the connection names. Enter the name of the cluster, as in Figure 6.
Next you must provide the name of the first queue manager in the cluster (the local queue manager), as shown in Figure 7. This queue manager must not be a member of another cluster, or a repository queue manager.
MQSeries Clustering on Windows

The next piece of information needed is the name of the other repository queue manager, and the connection name (in a TCP/IP environment, this is either the hostname or network address of the remote system) as shown in Figure 8.

**Figure 8**

The cluster creation wizard will continue with the naming of the cluster sender / receiver channels. First, the cluster receiver channels are defined. Enter the channel name and the local connection name as in Figure 9.
Figure 9

The remote cluster receiver channel is also defined at this point. Enter the receiver channel name and remote connection name as in figure 10. The cluster sender channels will be automatically created for you.

Figure 10
**MQSeries Clustering on Windows**

The wizard will display a list of the objects to be created, including the cluster sender channels. Remember, corresponding sender and receiver channels must be named the same, so the remote cluster sender channel is named the same as the local cluster receiver channel and vice-versa. Click next, and the wizard proceeds to create the cluster objects. You can also print out the names of these objects at this point. MQSeries will now attempt to contact the remote queue manager to create the remote objects, as shown in Figure 11.

![Figure 11](image)

Now, from MQSeries Explorer, you must right-click on the ‘Clusters’ folder and select ‘Show Cluster’. Enter the cluster name and the local repository queue manager name. The left pane of your MQSeries Explorer will now display the cluster queue managers and their MQSeries objects, as in Figure 12.

![Figure 12](image)
4.3 Creating and testing a distributed cluster queue

Next, we will create multiple definitions of a cluster queue called clq_test. The idea is to create a clq_test queue on each of the repository queue managers (cgilab71 and cgilab72) for load balancing purposes. It is important to familiarize yourself with the way MQSeries Explorer represents local and remote cluster queues. Local shared queues have the ‘Microsoft network share open hand’ icon attached to them, and when right-clicked, allow you to browse messages, put test messages, or share the queue in a cluster (or list of clusters). Figure 13 shows the right pane view from MQSeries Explorer, and depicts three queues. Two of the queues are called clq_test, but only the one with the ‘hand icon’ is the local shared cluster queue. The other clq_test queue is a remote cluster queue hosted by QM_cgilab71. The clq_test_notshared queue is a simple, local queue.

<table>
<thead>
<tr>
<th>Name</th>
<th>Queue Type</th>
<th>Cluster Name</th>
<th>Cluster Namelist</th>
<th>Hosting Queue Manager</th>
</tr>
</thead>
<tbody>
<tr>
<td>clq_test</td>
<td>Local</td>
<td>CSILAB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>clq_test</td>
<td>Cluster</td>
<td></td>
<td></td>
<td>QM_cgilab71</td>
</tr>
<tr>
<td>clq_test_notshared</td>
<td>Local</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 13

In order to create a cluster queue, first define a local queue. You may choose to share this queue in the cluster as you create the queue, or after it has been created. To create a local queue, right-click on the ‘Queues’ folder and select ‘New…Local Queue’ as show in Figure 14.

Figure 14

The Create local queue dialog box will start up and prompt for the queue attributes.
First, enter the queue name in the general tab dialog box, and then select the Cluster tab to share the queue in the newly created cluster, as shown in Figures 15 and 16. We will share the queue as we create it in this example. If you decided to share the queue later, you could right click on the queue and select ‘Share in Cluster’ to enter the cluster information.

Figure 15
At this point, the queue has been created on the local queue manager. However, for remote queue managers to see the queue, they must refresh their repository information first.

Notice that the default bind type of the cluster queue is set to OnOpen. This indicates that applications will be bound to this instance of the cluster queue at open time if they open the queue with default options. This guarantees that all messages put to the queue (while the application still has it open) will only be routed here. The alternative to this is Not Fixed, which places responsibility on the applications themselves to specify a bind type when they open a queue. This will be discussed in detail in a subsequent article.

The next step is to create another instance of the clq_test queue on the second repository queue manager (cgilab71). Following the same procedures above on the QM_cgilab71 queue manager, create another queue called clq_test. Then create another local queue on QM_cgilab72 called clq_test_notshared and do not share it in the cluster. The resulting view from QM_cgilab72’s MQSeries Explorer should look like Figure 17.

<table>
<thead>
<tr>
<th>Name</th>
<th>Queue Type</th>
<th>Cluster Name</th>
<th>Cluster Namelist</th>
<th>Hosting Queue Manager</th>
</tr>
</thead>
<tbody>
<tr>
<td>clq_test</td>
<td>Local</td>
<td>CGILAB</td>
<td></td>
<td>QM_cgilab71</td>
</tr>
<tr>
<td>clq_test</td>
<td>Cluster</td>
<td></td>
<td></td>
<td>QM_cgilab71</td>
</tr>
<tr>
<td>clq_test_notshared</td>
<td>Local</td>
<td></td>
<td></td>
<td>QM_cgilab71</td>
</tr>
</tbody>
</table>

We can now put test messages to the clq_test cluster queues. First, we will put a message to the clq_test queue shared by cgilab71, from QM_cgilab72 (the second queue shown in Figure 17). From the MQSeries Explorer window on cgilab72, right click on the clq_test queue and select “Put Test Message…” as shown in Figure 18. Enter “this is a test” in the message data, and click OK.
In order to view the message contents, you now need to use either the MQSeries Explorer on cgilab71, or use the MQSeries Explorer on cgilab72 and select the QM_cgilab71 queue manager from the ‘clusters’ folder. Figure 19 shows the view from the former. Right click on the clq_test queue and select “Browse Messages…”
Figure 19

The message data, as entered in then displayed as shown in Figure 20. Note that the columns displayed here are the default columns. You can customize this column display by clicking on the ‘Columns’ button.

![Message Browser](image)

**Message Browser**

Queue Manager Name: QM_cgilab71
Queue Name: cliq_test

<table>
<thead>
<tr>
<th>n</th>
<th>Put Date/Time</th>
<th>User Identifier</th>
<th>Put Application Name</th>
<th>Format</th>
<th>Data Length</th>
<th>Message Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1/12/00 3:18:37 PM</td>
<td>manadmin</td>
<td>D:\WINNT\System32\WMC.EXE</td>
<td>MQSTR</td>
<td>14</td>
<td>this is a test</td>
</tr>
</tbody>
</table>

All available messages on the queue have been browsed. Waiting for new messages...

Figure 20
4.4 Joining an existing cluster

Next, we’ll add cgilab221 to the CGILAB cluster. The first step is to create a stand-alone queue manager called QM_cgilab221, in the same way we did in section 3.1. Once the queue manager is created, all you need to do in order to make it a member of the cluster is the open MQSeries Explorer, right click on the QM_cgilab221 queue manager, and select “Join Cluster…” as shown in Figure 21. **Note: if you do not see this option, it means that your configuration does not support clustering. For example, if you use DHCP for dynamic IP addressing, you cannot become a cluster member.**

**Figure 21**

At this point, the **Join Cluster Wizard** starts up, and prompts for the cluster information. You must first enter the name of the cluster (CGILAB) as in Figure 22. Then, provide the name of one of the cluster queue managers that is acting as a repository. In our case, both QM_cgilab71 and QM_cgilab72 are repository queue managers, and either could be selected. We will use QM_cgilab72. You must also provide the connection name for this queue manager; in our case it is cgilab72.on.bell.ca (see Figure 23).
Figure 22

Join Cluster Wizard

1. Identify the cluster

Enter the name of the cluster you want your queue manager 'QM_cglab221' to join.

Cluster

Figure 23

Join Cluster Wizard

2. Identify the repository queue manager

Identify a repository queue manager within the cluster 'CGLAB'. This queue manager must already exist, and be a repository.

Select where the queue manager is located:
- Local (on this computer)
- Remote (on another computer)

Enter the queue manager's name:
QM_cglab72

Enter the queue manager's connection name:
cglab72.on.bell.ca

This is usually the same as the queue manager's host name.
As with creating a cluster (section 3.2), the cluster sender/receiver channels must also be named. First, the new queue manager’s receiving channel must be identified. Following our standard naming convention, the receiver channel would be `TO_cgilab221` and the connection name would be `cgilab221.on.bell.ca`. Figure 24 shows this step.

![Join Cluster Wizard diagram](image)

**Figure 24**

Next, provide the names of the repository queue manager’s cluster receiver channel as in Figure 25.
MQSeries Clustering on Windows

Figure 25

Click Finish to complete the configuration, and MQSeries will attempt to connect the queue manager to the cluster. Although everything will appear to work, you still need to display the cluster by right clicking on the Cluster folder and selecting Show Cluster. You must provide the cluster name, a repository queue manager name, and the associated connection name. (See Figures 26 and 27).

Figure 26
Now, the entire cluster containing all 3 queue managers can be displayed and managed from a single node, as shown in Figure 28.

**Figure 27**

You can see how simplified the management of MQSeries becomes using clustering. In addition to creating the cluster, adding cluster queue managers, and adding cluster queues, you can now modify and manage attributes of all these objects from a single location.

**Figure 28**