



# MQ Clustering

## Creating an MQ Cluster on Windows

**Revision:**

1.0

**Prepared by:**

Scott Meridew – President  
MQ Squared Inc.

IBM Certified System Administrator – WebSphere MQ  
IBM Certified Solution Designer – WebSphere MQ  
IBM Certified Solution Developer – WebSphere MQ

[www.mqsquared.ca](http://www.mqsquared.ca)

[scott@mqsquared.ca](mailto:scott@mqsquared.ca)

416.846.9939





---

---

**TABLE OF CONTENTS**

<b>1</b>	<b>CONVENTIONS</b> .....	<b>3</b>
<b>2</b>	<b>AUDIENCE</b> .....	<b>3</b>
<b>3</b>	<b>CREATING AN MQ CLUSTER</b> .....	<b>4</b>
<b>4</b>	<b>DEFINING THE CLUSTER USING MQ EXPLORER</b> .....	<b>5</b>
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

---



---

## 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*
- MQ 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 MQ developers who are implementing an MQ cluster, as well as system architects and designers.



---

### 3 Creating an MQ cluster

With the introduction of MQ for Windows V5, IBM has included graphical utilities for the management of MQ. However, since MQ 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 MQ for Windows V5 library does not have any GUI screen shots showing how to build or manage an MQ cluster. This document will guide you through the configuration and management of an MQ cluster using the graphical tools provided in MQ for Windows.

The graphical tools provided with MQ Explorer for Windows V5 include:

- *MQ Explorer* – a Microsoft Management Console snap-in that allows an administrator to add/remove/manage MQ objects (both local and remote).
- *MQ Services* – a Microsoft Management Console snap-in that allows an administrator to add/remote/manage MQ services such as listeners, traces, trigger monitors, and channel initiators.
- *MQ 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 MQ cluster on Windows; 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. 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 MQ first steps *default configuration wizard*, you can allow MQ to define your MQ objects, as it poses as series of questions about your environment. The trouble with using the default configuration is that the MQ 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 MQ 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 Windows cluster.**



---

## 4 Defining the cluster using MQ 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 Windows 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 MQ 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 MQ Explorer tool. Open the program up by selecting Start...Programs...IBM MQ...MQ Explorer. Then right click on Queue Managers, and select New as show in Figure 1.



## Creating an MQ Cluster

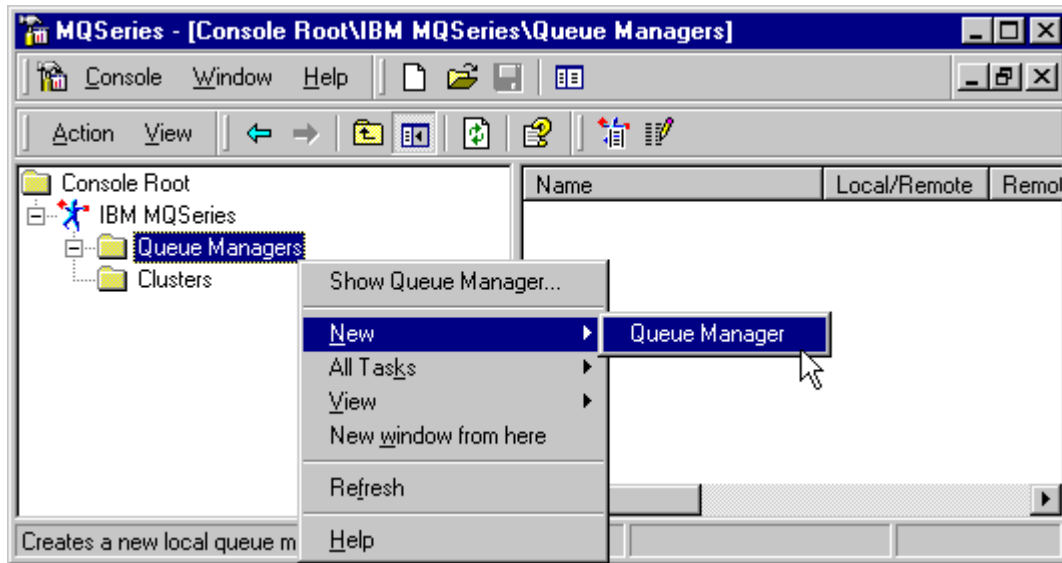


Figure 1



## Creating an MQ Cluster

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.

**Create Queue Manager (Step 1)**

Queue Manager: QM\_cgilab72

Make this the default queue manager

Def. Transmission Queue:

Dead Letter Queue:

Max Handle Limit: 256

Trigger Interval: 999999999

Max Uncommitted Msgs: 1000

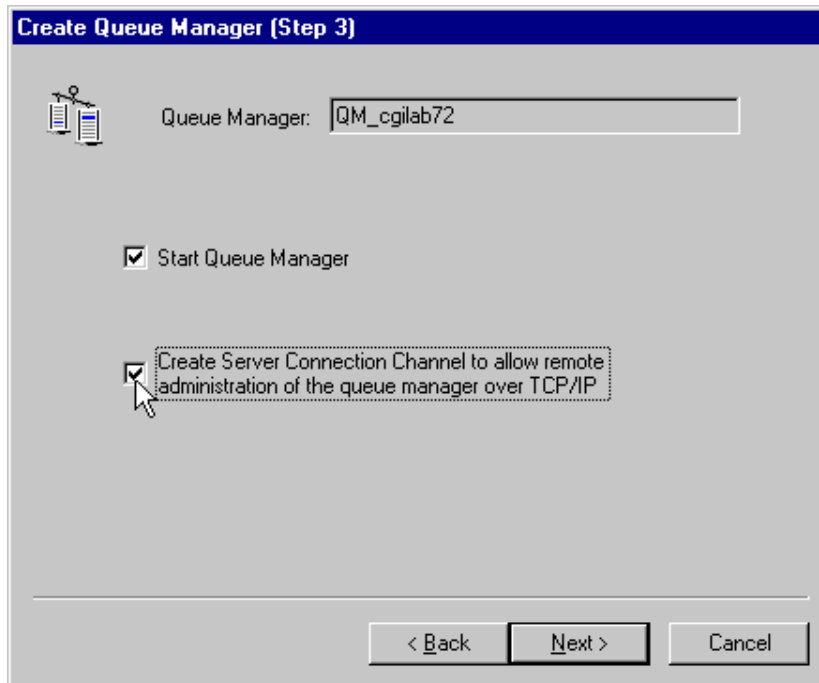
< Back   Next >   Cancel

**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 MQ Explorer window. Figure 3 shows this dialog box.

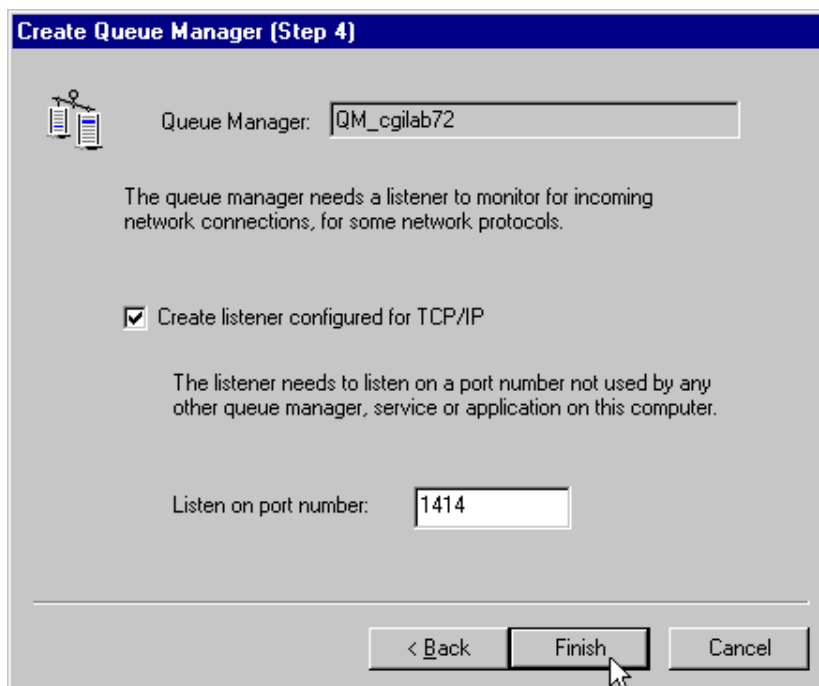


## Creating an MQ Cluster



**Figure 3**

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



**Figure 4**





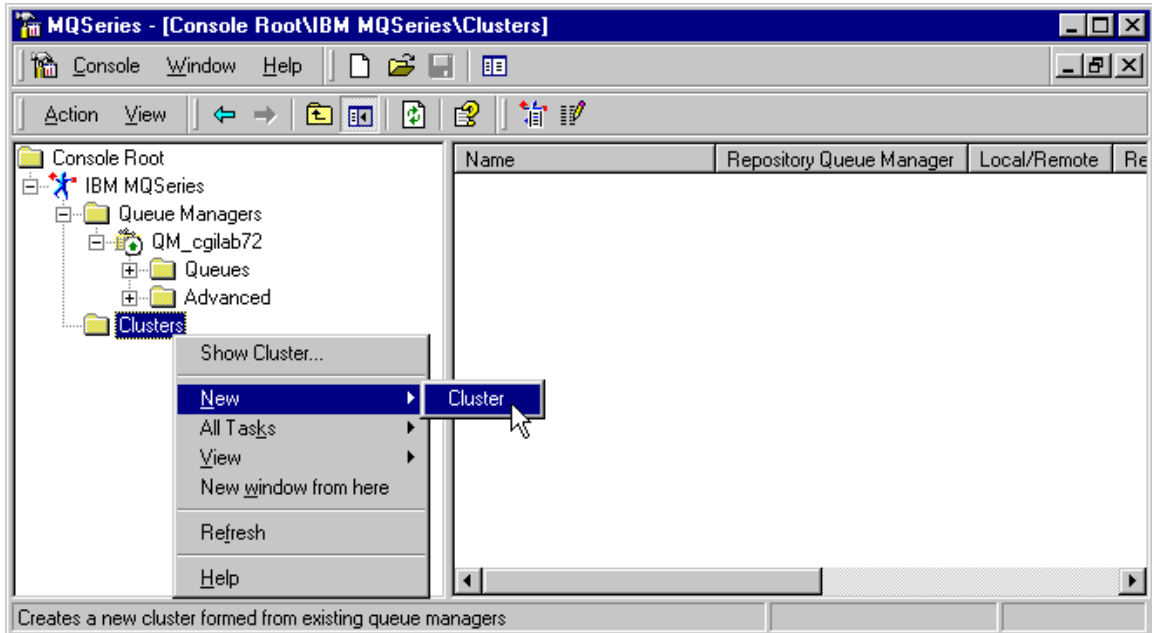
## *Creating an MQ Cluster*

**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 MQ 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.

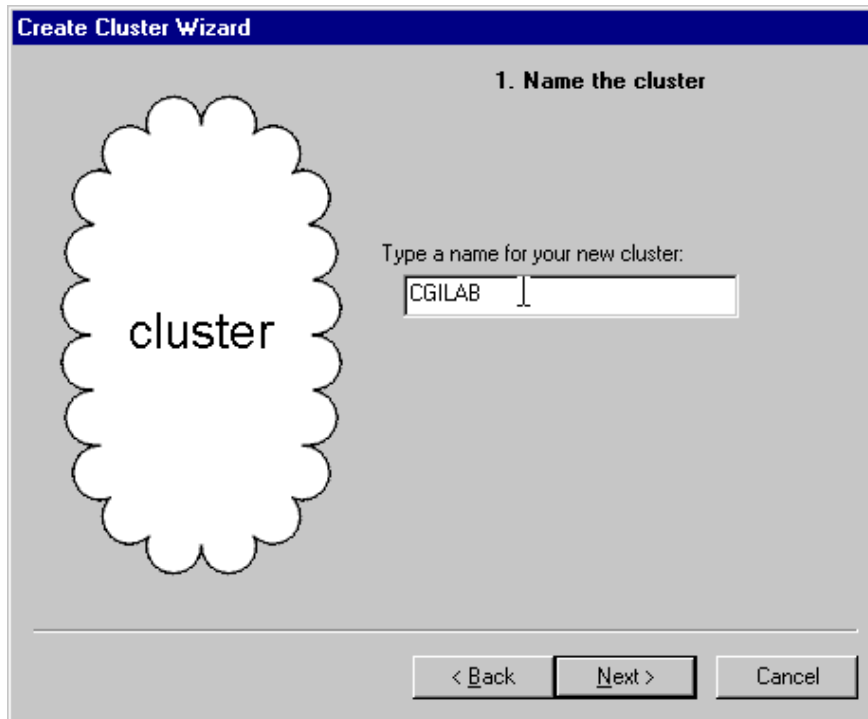


Figure 6

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.



Figure 7



## Creating an MQ Cluster

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.

**Create Cluster Wizard**

**3. Choose the second repository queue manager**

Choose the second queue manager that will be a repository in your cluster 'CGILAB'. It must not already be a repository queue manager, or a member of another cluster.

Select where the queue manager is located:

Local (on this computer)

Remote (on another computer)

Enter the queue manager's name:

QM\_cgilab71

Enter the queue manager's connection name:

cgilab71.on.bell.ca

This is usually the same as the queue manager's hostname.

< Back    Next >    Cancel

**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.



## Creating an MQ Cluster

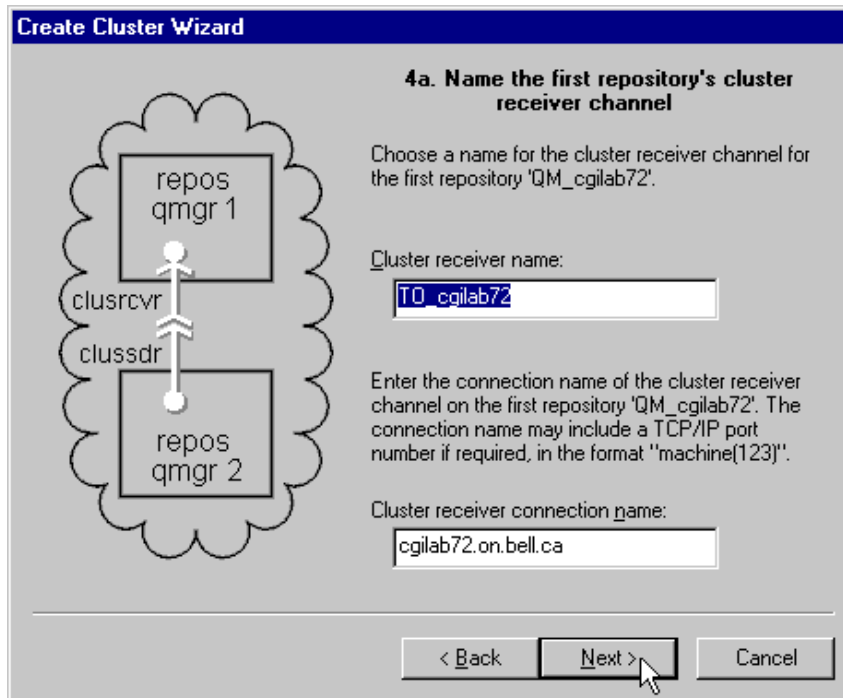


Figure 9

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.

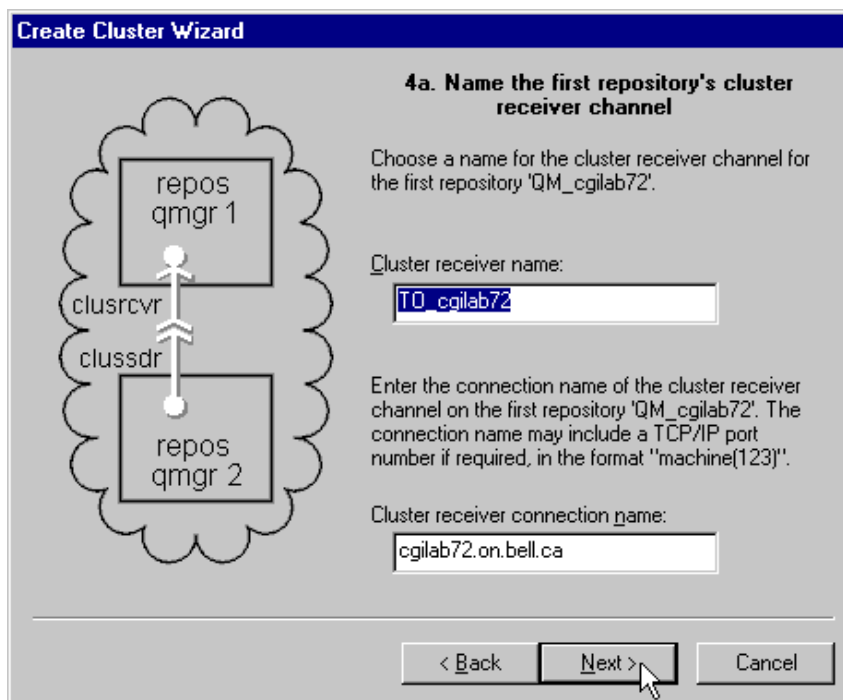


Figure 10



## Creating an MQ Cluster

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. MQ will now attempt to contact the remote queue manager to create the remote objects, as shown in Figure 11.

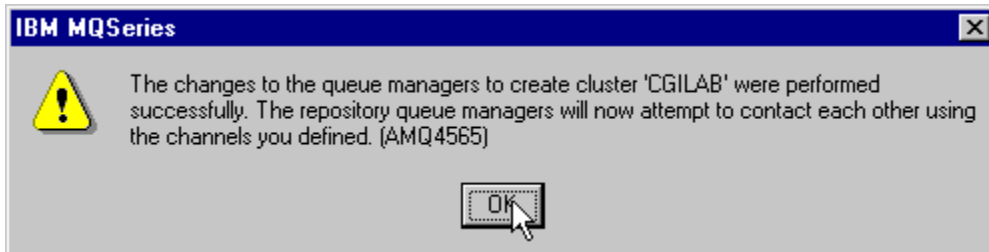


Figure 11

Now, from MQ 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 MQ Explorer will now display the cluster queue managers and their MQ objects, as in Figure 12.

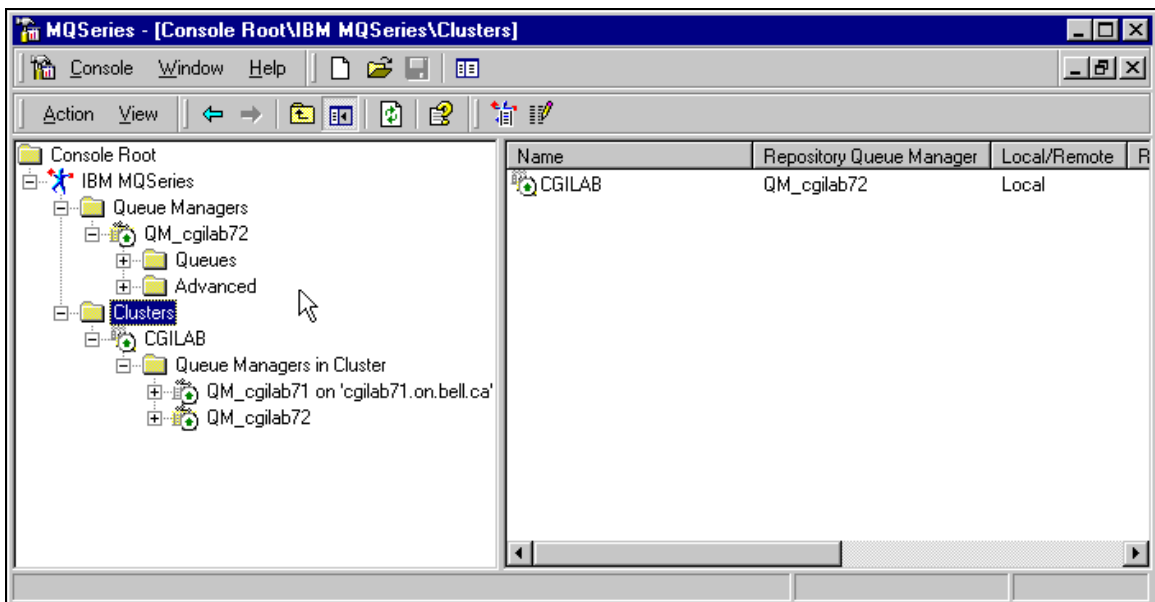


Figure 12



## *Creating an MQ Cluster*



### 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 MQ 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 MQ 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.

Name	Queue Type	Cluster Name	Cluster Namelist	Hosting Queue Manager
clq_test	Local	CGILAB		
clq_test	Cluster			QM_cgilab71
clq_test_notshared	Local			

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.

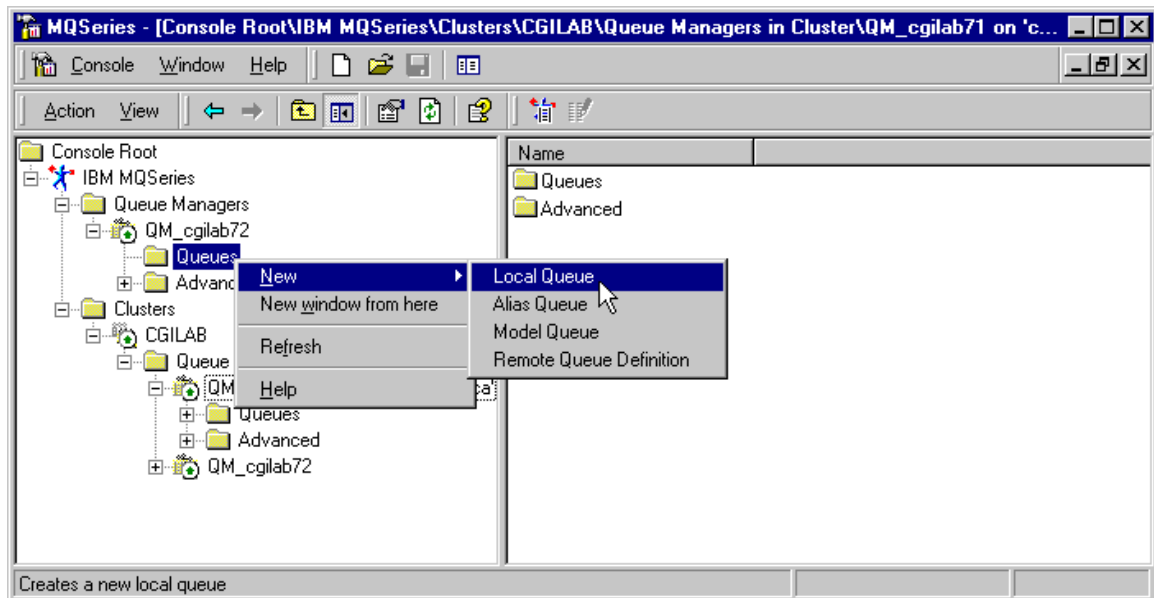


Figure 14

The Create local queue dialog box will start up and prompt for the queue attributes.





## Creating an MQ Cluster

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.

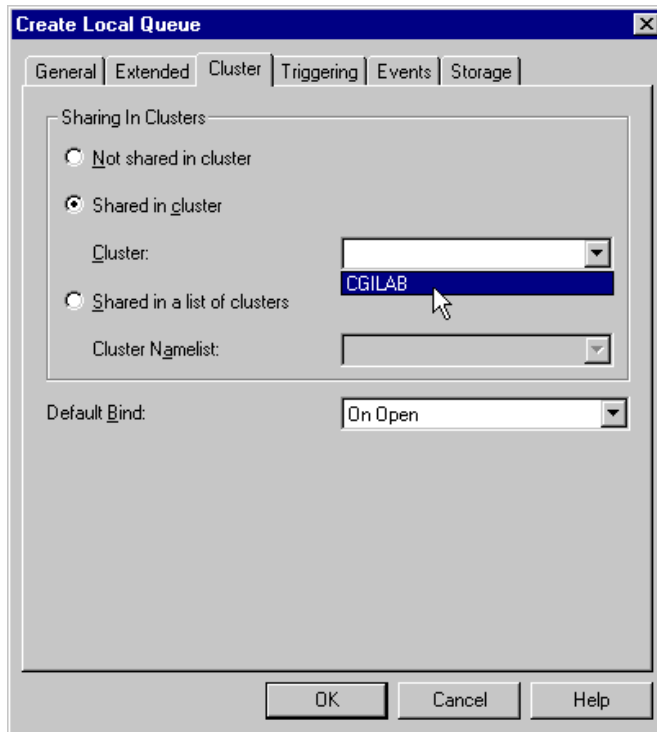
The screenshot shows the 'Create Local Queue' dialog box with the 'Cluster' tab selected. The dialog box has a title bar with 'Create Local Queue' and a close button. Below the title bar are tabs for 'General', 'Extended', 'Cluster', 'Triggering', 'Events', and 'Storage'. The 'Cluster' tab is active, and a mouse cursor is pointing at it. The queue name 'clq\_test' is entered in the 'Queue Name' field. The 'Type' is set to 'Local'. The 'Description' field is empty. The 'Put Messages' and 'Get Messages' options are both set to 'Allowed'. The 'Default Priority' is set to '0'. The 'Default Persistence' is set to 'Not Persistent'. The 'Scope' is set to 'Queue Manager'. The 'Usage' is set to 'Normal'. At the bottom of the dialog box are three buttons: 'OK', 'Cancel', and 'Help'.

Field	Value
Queue Name:	clq_test
Type:	Local
Description:	
Put Messages:	Allowed
Get Messages:	Allowed
Default Priority:	0
Default Persistence:	Not Persistent
Scope:	Queue Manager
Usage:	Normal

Figure 15



## Creating an MQ Cluster



**Figure 16**

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 MQ Explorer should look like Figure 17.

Name	Queue Type	Cluster Name	Cluster Namelist	Hosting Queue Manager
clq_test	Local	CGILAB		
clq_test	Cluster			QM_cgilab71
clq_test_notshared	Local			

**Figure 17**

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 MQ 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.



## Creating an MQ Cluster

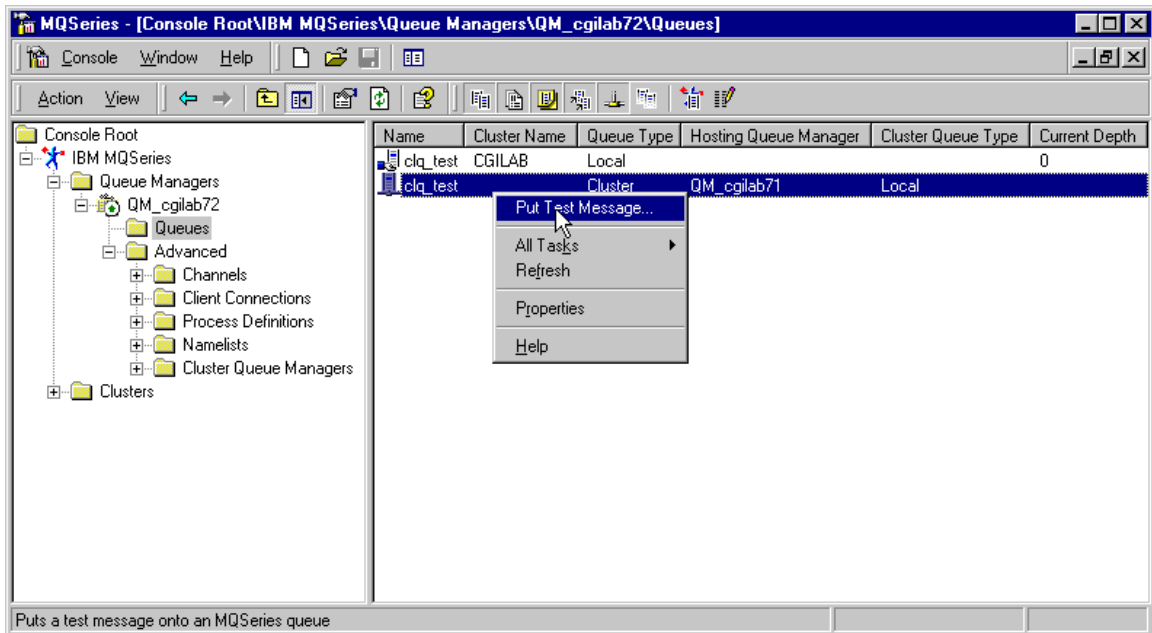
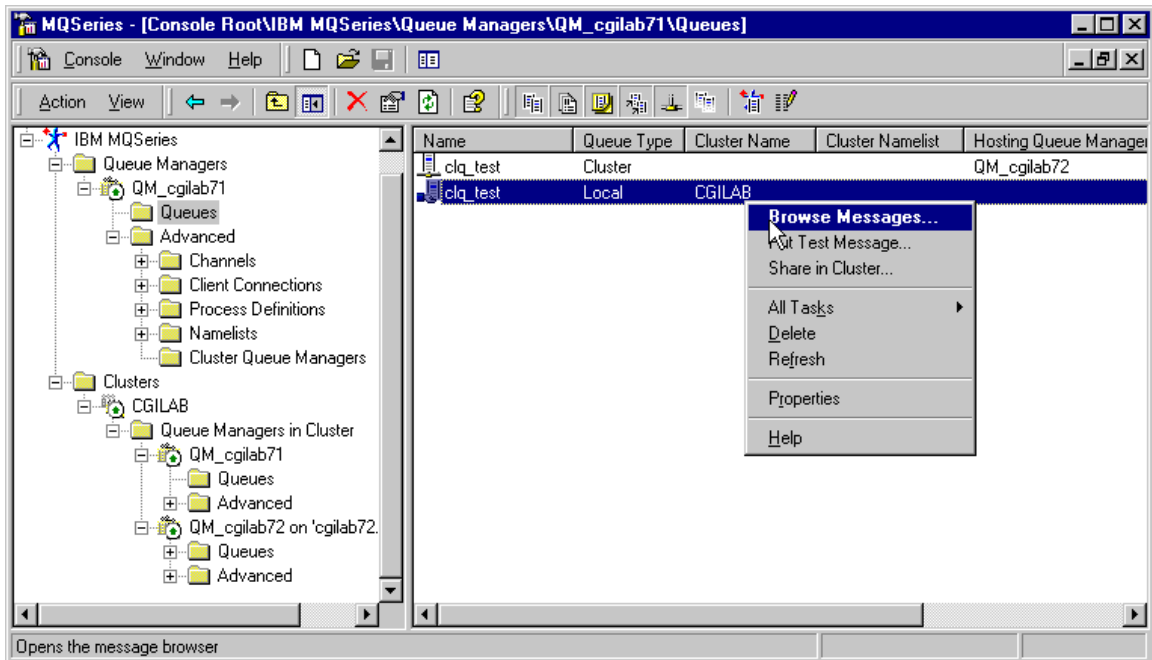


Figure 18

In order to view the message contents, you now need to use either the MQ Explorer on cgilab71, or use the MQ 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..."

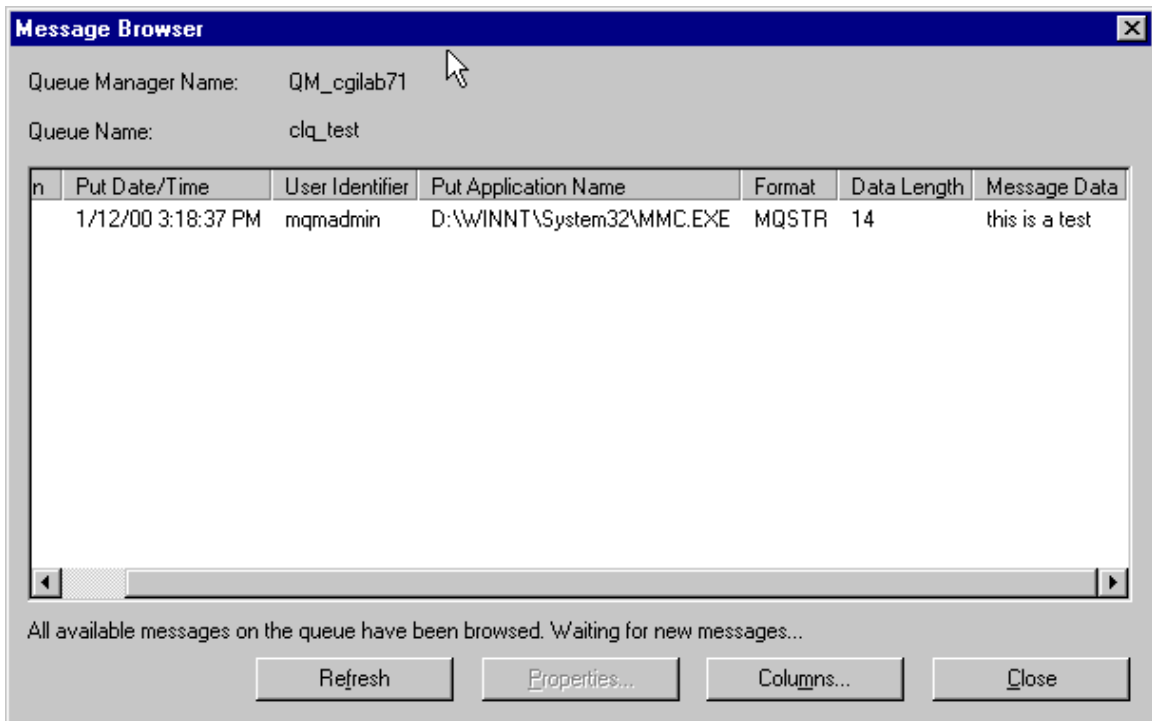


## Creating an MQ Cluster



**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.



**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 MQ 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.**

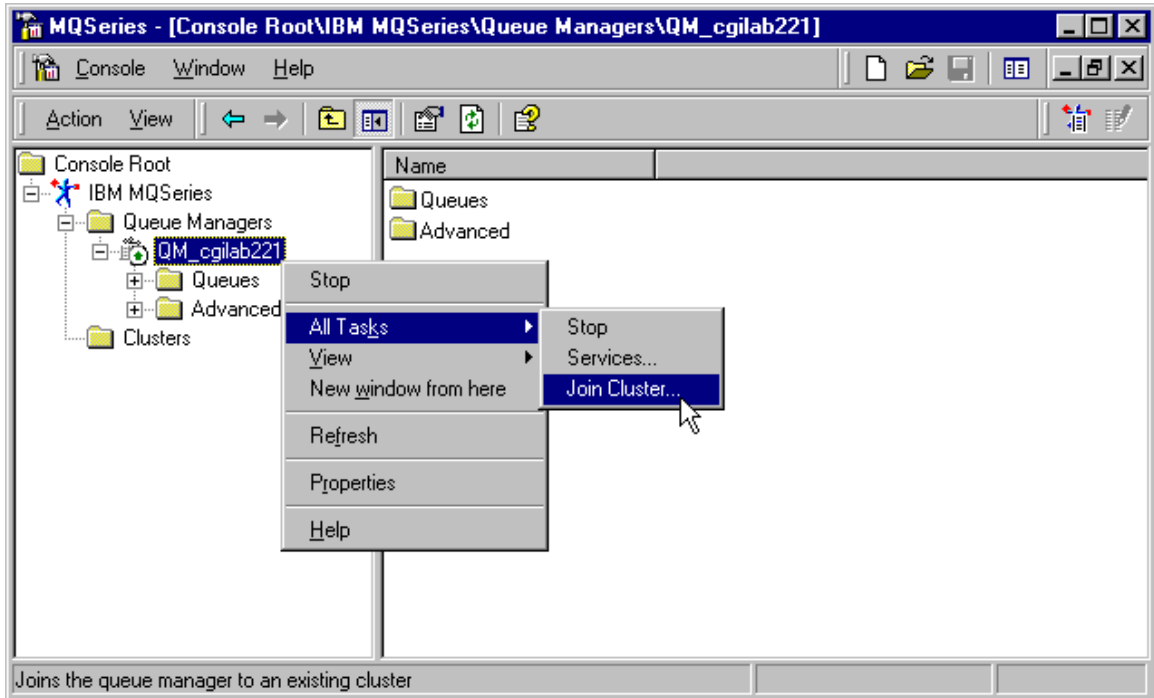


Figure 21

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).



## Creating an MQ Cluster

The screenshot shows a window titled "Join Cluster Wizard" with a close button in the top right corner. The main heading is "1. Identify the cluster". On the left, there is a diagram with a box labeled "qmgr" and a cloud-like shape labeled "cluster". The text on the right says: "Enter the name of the cluster you want your queue manager 'QM\_cgilab221' to join:". Below this text is a text input field containing "CGILAB". At the bottom, there are three buttons: "< Back", "Next >", and "Cancel". A mouse cursor is pointing at the "Next >" button.

Figure 22

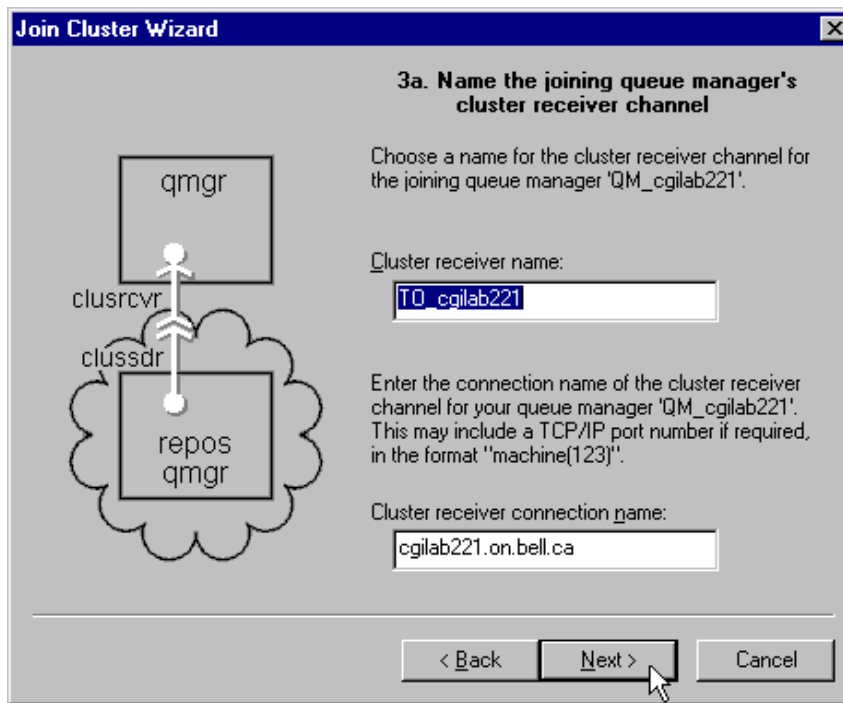
The screenshot shows a window titled "Join Cluster Wizard" with a close button in the top right corner. The main heading is "2. Identify the repository queue manager". On the left, there is a diagram with a box labeled "qmgr" and a cloud-like shape labeled "repos qmgr". The text on the right says: "Identify a repository queue manager within the cluster 'CGILAB'. This queue manager must already exist, and be a repository." Below this text, it says: "Select where the queue manager is located:" followed by two radio button options: "Local (on this computer)" and "Remote (on another computer)". The "Remote" option is selected. Below the radio buttons, it says: "Enter the queue manager's name:" followed by a text input field containing "QM\_cgilab72". Below that, it says: "Enter the queue manager's connection name:" followed by a text input field containing "cgilab72.on.bell.ca". A note below the input fields says: "This is usually the same as the queue manager's hostname." At the bottom, there are three buttons: "< Back", "Next >", and "Cancel". A mouse cursor is pointing at the "Next >" button.

Figure 23



## Creating an MQ Cluster

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.

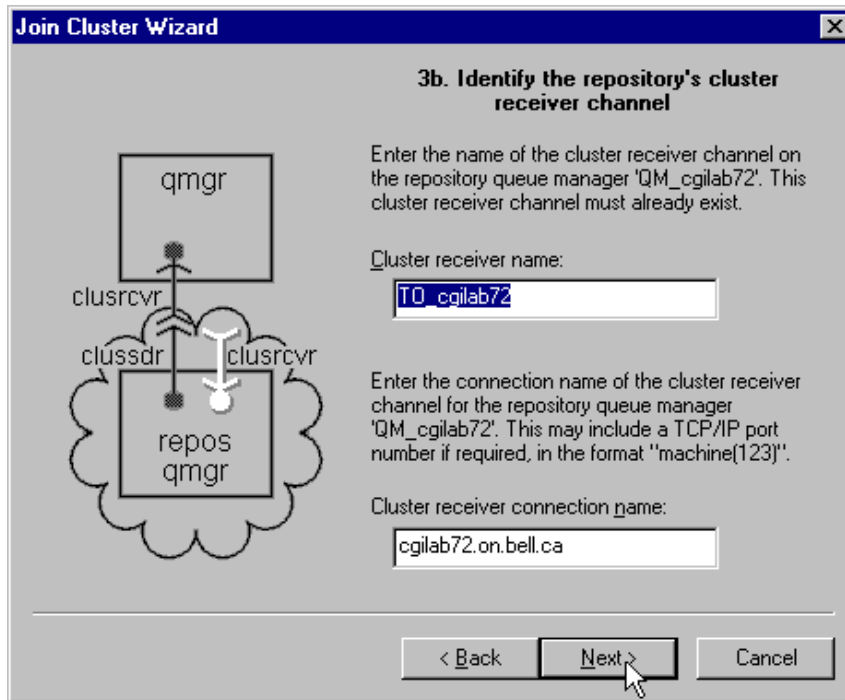


**Figure 24**

Next, provide the names of the repository queue manager's cluster receiver channel as in Figure 25.

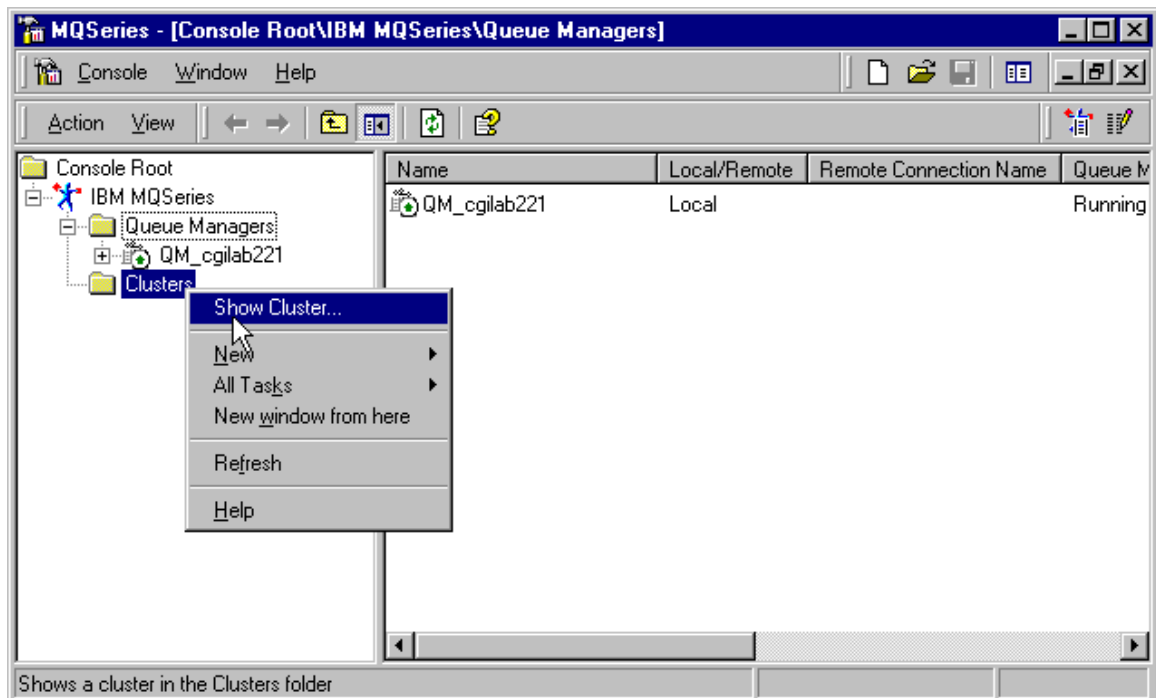


## Creating an MQ Cluster



**Figure 25**

Click Finish to complete the configuration, and MQ 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**





## Creating an MQ Cluster

**Show Cluster**

Enter the name of a cluster and the name and connection information for one of its repository queue managers.

Cluster Name:

Repository queue manager is local

Queue Manager Name:

Repository queue manager is remote

Queue Manager Name:

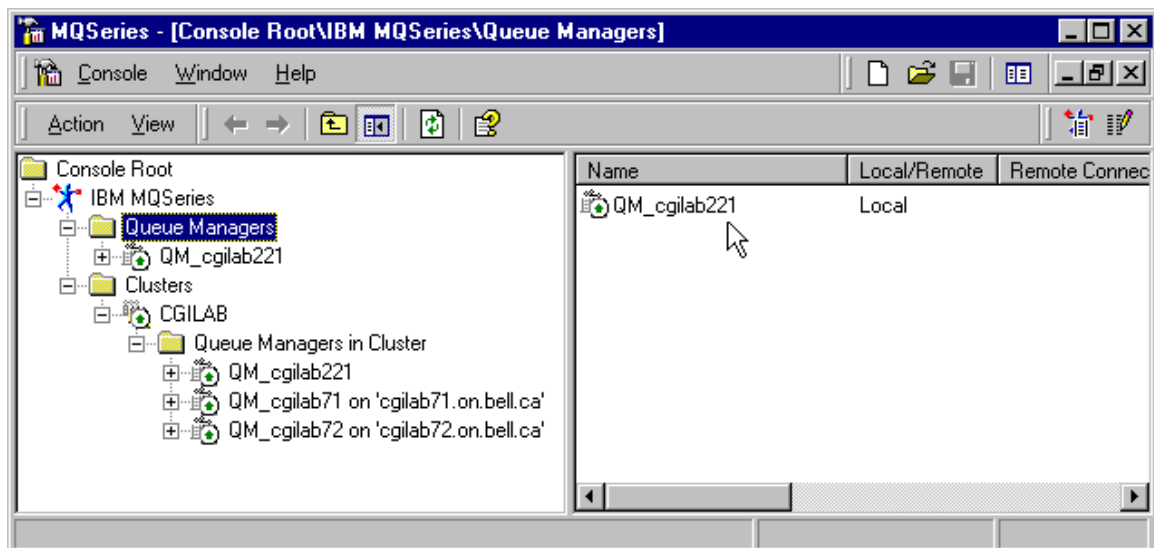
Connection Name:

This is usually the same as the repository queue manager's hostname.

OK Cancel Help

**Figure 27**

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



**Figure 28**

You can see how simplified the management of MQ 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.