

Define local queues to the cluster

- Specify the CLUSTER keyword on the local queue definition
 - Causes the queue to be advertised to the cluster
 - Available to other QMGRs in the cluster
 - No need for remote queue definition
- Example:

```
DEFINE QLOCAL(MYQUEUE1) CLUSTER(MYCLUSTER)
```

This defines the local queue MYQUEUE1 to the cluster MYCLUSTER

© Copyright IBM Corporation 2008

Figure 11-25. Define local queues to the cluster

WM203 / VM2032.0

Notes:

Verify and test the cluster

- Issue display command to verify the cluster:

```
display clusqmgr(*)
```

- And the results if issued from MYQMGR2:

```
1 : dis clusqmgr(*)  
AMQ8441: Display Cluster Queue Manager details.  
    CLUSQMGR(MYQMGR2) CLUSTER(MYCLUSTER) CHANNEL(TO.MYQMGR2)  
AMQ8441: Display Cluster Queue Manager details.  
    CLUSQMGR(MYQMGR1) CLUSTER(MYCLUSTER) CHANNEL(TO.MYQMGR1)
```

- Test using sample programs amqsput and amqsget

© Copyright IBM Corporation 2008

Figure 11-26. Verify and test the cluster

WM203 / VM2032.0

Notes:

Display channel status

- To check the channel definitions, issue the command:

```
display chstatus(*)
```

- And the resulting display is like:

```
1 : dis chstatus(*)
AMQ8417: Display Channel Status details.
  CHANNEL(TO.MYQMGR2)          XMITQ( )
  CONNAME(9.20.40.24)          CURRENT
  CHLTYPE(CLUSRCVR)           STATUS(RUNNING)
  RQMNAME(MYQMGR1)

AMQ8417: Display Channel Status details.
  CHANNEL(TO.MYQMGR1)          XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
  CONNAME(9.20.51.25)          CURRENT
  CHLTYPE(CLUSSDR)             STATUS(RUNNING)
  RQMNAME(MYQMGR1)
```

© Copyright IBM Corporation 2008

Figure 11-27. Display channel status

WM203 / VM2032.0

Notes:

Because the MYQUEUE1 queue has been advertised to the cluster there is no need for remote-queue definitions. Applications running on MYQMGR2 and applications running on MYQMGR1 can put messages to the MYQUEUE1 queue. They can receive responses to their messages by providing a reply-to queue and specifying its name when they put messages.

Cluster setup topic summary

Having completed this topic, you should be able to:

- Outline the steps required to setup a basic cluster
- Define the cluster related queue manager attributes
- List the necessary queue and channel definitions

© Copyright IBM Corporation 2008

Figure 11-28. Cluster setup topic summary

WM203 / VM2032.0

Notes:

11.3. Cluster administration

Cluster administration topic objectives

After completing this topic, you should be able to:

- List the commands used to control clusters
- Describe the administration tasks to consider in a cluster environment

© Copyright IBM Corporation 2008

Figure 11-29. Cluster administration topic objectives

WM203 / VM2032.0

Notes:

MQ commands for controlling clusters

- DISPLAY CLUSQMGR
 - Cluster information
- SUSPEND QMGR
 - Remove a QMGR from a cluster
- RESUME QMGR
 - Re-instate a QMGR to a cluster
- REFRESH CLUSTER
 - Discard locally held data about a cluster
 - Objects are rebuilt according to REPOS parm
 - Enforces a COLD START of the queue manager
 - Causes temporary disruption for pub/sub traffic
- RESET CLUSTER
 - Issued by a repository queue manager
 - Forcibly removes a QMGR from a cluster
 - Only used in exceptional circumstances

© Copyright IBM Corporation 2008

Figure 11-30. MQ commands for controlling clusters

WM203 / VM2032.0

Notes:

- **DISPLAY CLUSQMGR**
 - If you issue this command from a queue manager with a full repository, the information returned pertains to every queue manager in the cluster. If you issue this command from a queue manager that does not have a full repository, the information returned pertains only to the queue managers in which it has an interest.
- **SUSPEND QMGR and RESUME QMGR**
 - Use the SUSPEND QMGR command and RESUME QMGR command to remove a queue manager from a cluster temporarily, for example for maintenance, and then to reinstate it using the RESUME command. While suspended, the workload management exit does not route any messages to this queue manager.
- **REFRESH CLUSTER**
 - You can issue REFRESH CLUSTER(*) to refresh the queue manager in all of the clusters it is a member of. If used with REPOS(YES) this command has the additional effect of forcing the queue manager to restart its search for full

repositories from the information in the local CLUSSDR definitions, even if the CLUSSDR connects the queue manager to several clusters.

- Using REFRESH CLUSTER(clustername) REPOS(NO) provides the default behavior. The queue manager retains knowledge of all cluster queue manager and cluster queues marked as locally defined and all cluster queue managers that are marked as full repositories.
- Using REFRESH CLUSTER(clustername) REPOS(YES) specifies that in addition to the default behavior, objects representing full repository cluster queue managers are also refreshed. This option may not be used if the queue manager is itself a full repository.

- **RESET CLUSTER**

- In an emergency where a queue manager is temporarily damaged, you might want to inform the rest of the cluster before the other queue managers try to send it messages. RESET CLUSTER can be used to remove the damaged queue manager. Later when the damaged queue manager is working again, you can use the REFRESH CLUSTER command to reverse the effect of RESET CLUSTER and put it back in the cluster again.
- Only use this command on advice from support personnel if unsure what the effect of using this command is. See the *Queue Manager Clusters* guide for more information
- The PCF equivalents to these commands are:

```
MQCMD_INQUIRE_CLUSTER_Q_MGR  
MQCMD_SUSPEND_Q_MGR_CLUSTER  
MQCMD_RESUME_Q_MGR_CLUSTER  
MQCMD_REFRESH_CLUSTER  
MQCMD_RESET_CLUSTER
```

Displaying information about cluster channels

```
DISPLAY CLUSQMGR (*) CLUSTER(name)
```

returns :

| | |
|--------------------|--|
| CLUSDATE | Date and time when the definition became available to the local queue manager |
| CLUSTIME | |
| DEFTYPE | How the cluster queue manager was defined |
| QMTYPE | Function of the queue manager in the cluster; whether it provides full or partial repository service |
| QMID | Internally generated unique queue manager name |
| STATUS | Current status of the channel for this queue manager; may be any valid channel status |
| SUSPEND CHANNEL | 'Yes' or 'No' as a result of a SUSPEND QMGR command The cluster-receiver channel name for the queue manager |

- Additionally, channel attributes are returned for all or the (generically) named channels

© Copyright IBM Corporation 2008

Figure 11-31. Displaying information about cluster channels

WM203 / VM2032.0

Notes:

The DISPLAY CLUSMGR command returns cluster information about queue managers in a cluster which is stored in the local SYSTEM.CLUSTER.REPOSITORY.QUEUE.

Definition Type may be:

- **CLUSSDR** - as a cluster-sender channel from an explicit definition
- **CLUSSDRA** - as a cluster-sender by auto-definition
- **CLUSSDRB** - as a cluster-sender channel, both from an explicit definition and by auto-definition
- **CLUSRCVR** - as a cluster-receiver channel.

Format of the cluster commands

```
SUSPEND QMGR CLUSTER(clustername)
    MODE (FORCE | QUIESCE)

RESUME QMGR CLUSTER(clustername)

REFRESH CLUSTER(clustername)
    REPOS (YES | NO)

RESET CLUSTER(clustername)
    QMNAME (qmname) /QMID (qmid)
    ACTION (FORCEREMOVE) QUEUES (YES | NO)
```

© Copyright IBM Corporation 2008

Figure 11-32. Format of the cluster commands

WM203 / VM2032.0

Notes:

- **REFRESH CLUSTER(*clustername*) REPOS(YES)** - Default action for the queue manager is to retain a knowledge of all cluster queue managers and cluster queues marked as locally defined, and all cluster queue managers that are full repository. If a full repository it retains knowledge of the other cluster queue managers in the cluster.
- **REFRESH CLUSTER (*clustername*) REPOS(NO)** - In addition to default action objects represent full repository queue cluster queue managers are also refreshed. You cannot use this option if the queue manager is a full repository.
- **REFRESH CLUSTER(*)** refreshes the queue manager in all of the clusters it is a member of.
- **RESET CLUSTER (*clustername*) QMNAME(*qmname*) ACTION(FORCEREMOVE) QUEUES(NO)** or the command **RESET CLUSTER (*clustername*) QMID(*qmid*) ACTION(FORCEREMOVE) QUEUES(NO)** - You cannot specify both QMNAME and QMID.

- **RESET CLUSTER (clustername) QMID(qmid) ACTION(FORCEREMOVE) QUEUES(NO)** - You cannot specify both QMNAME and QMID. This command can only be issued from the full repository queue manager. Use QMID when there is more than one queue manager in the cluster with the name QMNAME and you want to forcibly remove only one queue manager from the cluster. Do not use this command as a shortcut to remove a queue manager from a cluster.
- **RESET CLUSTER(clustername) QUEUES(YES)** - Means that any references to a cluster queue or queues owned by the queue manager being force-removed are removed from the cluster in addition to the cluster queue manager itself.

Cluster commands

| PCF Command | MQSC | Control Command | IBM WebSphere For MQ Explorer |
|-------------------------------|------------------|-----------------|-------------------------------|
| Inquire Cluster Queue Manager | DISPLAY CLUSQMGR | no equivalent | YES |
| Refresh Cluster | REFRESH CLUSTER | no equivalent | YES |
| Reset cluster | RESET CLUSTER | no equivalent | NO |
| Resume Queue Manager Cluster | RESUME CLUSTER | no equivalent | YES |
| Suspend Queue Manager Cluster | SUSPEND CLUSTER | no equivalent | YES |

© Copyright IBM Corporation 2008

Figure 11-33. Cluster commands

WM203 / VM2032.0

Notes:

Cluster administration considerations

- Maintaining a queue manager
 - SUSPEND and RESUME a QMGR
- Refreshing a queue manager
 - REFRESH cluster command
- Recovering a queue manager
 - Restore the QMGR from a linear log
- Maintaining the cluster transmission queue
 - Avoid queue full or disk full conditions
 - Ensure PUT(enabled) and GET(enabled) always
- When a queue manager fails
 - undelivered messages backed out to the cluster XMITQ on the sending queue manager
- Cluster channels
 - Online monitoring using DISPLAY CHSTATUS

© Copyright IBM Corporation 2008

Figure 11-34. Cluster administration considerations

WM203 / VM2032.0

Notes:

- From time to time, you might need to perform maintenance on a queue manager that is part of a cluster. For example, you might need to take backups of the data in its queues, or apply fixes to the software. If the queue manager hosts any queues, its activities must be suspended. When the maintenance is complete, its activities can be resumed.
- While the queue manager is suspended the workload management routines avoid sending messages to it, other than messages that **have** to be handled by that queue manager. Messages that **have** to be handled by that queue manager include messages sent by the local queue manager.
- Effectively, the REFRESH CLUSTER command allows a queue manager to be *restarted* with respect to its full repository content. (WebSphere MQ ensures that no data is lost from your queues.)
- If you have to restore from a point-in-time backup, issue the REFRESH CLUSTER command on the restored queue manager for all clusters in which the queue manager participates.

- The availability and performance of the cluster transmission queue are essential to the performance of clusters. Make sure that it does not become full, and take care not to accidentally issue an ALTER command to set it either get-disabled or put-disabled.

Implementation recommendations

- Be clear about your requirements
 - Reduced system administration?
 - Workload balancing?
- Familiarize yourself with the queue manager clusters manual
 - Also MQSC command reference
- Naming conventions
 - Consistency is the key
- Document processes for system changes
 - Add and remove queue manager to/from cluster
 - Take queue manager offline for maintenance
- Have two full repositories
 - Consider where they are hosted
- Monitor health of the system queues and channels
- Take care in defining cluster topics in existing clusters
 - For publish/subscribe in V7

© Copyright IBM Corporation 2008

Figure 11-35. Implementation recommendations

WM203 / VM2032.0

Notes:

Cluster administration topic summary

Having completed this topic, you should be able to:

- List the commands used to control cluster
- Describe the administration tasks to consider in a cluster environment

© Copyright IBM Corporation 2008

Figure 11-36. Cluster administration topic summary

WM203 / VM2032.0

Notes:

11.4. Workload management

Workload management topic objectives

After completing this topic, you should be able to:

- Describe how clusters assist with workload management
- List the attributes that affect the workload balancing algorithm
- Explain how application considerations play a role in workload balancing

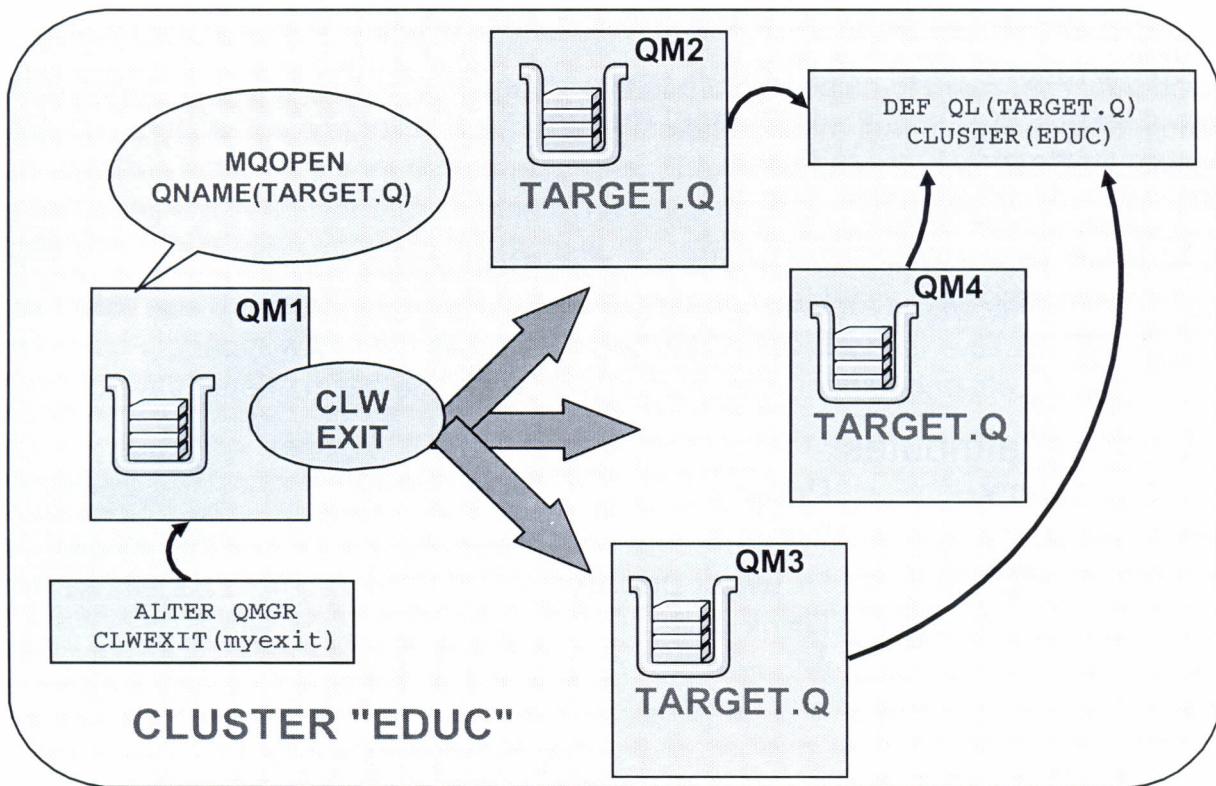
© Copyright IBM Corporation 2008

Figure 11-37. Workload management topic objectives

WM203 / VM2032.0

Notes:

Workload balancing



© Copyright IBM Corporation 2008

Figure 11-38. Workload balancing

WM203 / VM2032.0

Notes:

- Cluster support allows more than one queue manager to host an occurrence of the same queue. Thus, two or more queue managers can be clones of each other, capable of running the same applications and having local definitions of the same queues. There may be two reasons for doing implementing queues using this technique:
 - To increase the capacity available to process messages.
 - To introduce an ability to failover work from one server to another and improve availability of the service.
- This architecture can be used to spread the workload between queue managers, *if applications allow it to do so*.
- A built-in workload management algorithm determines the remote queue manager if there are multiple choices, based on availability and channel priorities. A local occurrence takes precedence by default.
- You may supply your own cluster workload exit and activate it by use of the ALTER QMGR command, as shown in the figure.

Workload balancing attributes

- Queue attributes

| | |
|----------|-----------------|
| CLWLPRTY | Priority |
| CLWLUSEQ | Use local queue |
| CLWLRANK | Rank |

- Queue manager attributes

| | |
|----------|------------------------|
| CLWLUSEQ | Use local queue |
| CLWLMRUC | Recently used channels |

- Channel attributes

| | |
|----------|------------------|
| CLWLPRTY | Priority |
| CLWLWGHT | Weight |
| NETPRTY | Network priority |
| CLWLRANK | Rank |

© Copyright IBM Corporation 2008

Figure 11-39. Workload balancing attributes

WM203 / VM2032.0

Notes:

- **CLWLPRTY**, the cluster workload priority attribute, determines the priority of the queue. Use this attribute to ensure that WebSphere MQ selects some destination queue managers in preference to others with a lower priority. WebSphere MQ selects the destinations with the highest priority before selecting destinations with the lowest cluster destination sequence number (or the most recently used one). Where there are two possible destinations, this attribute can be used to allow one queue manager to act as a failover, if the other queue manager becomes unavailable. In this situation, messages go to the highest priority queue manager until it becomes unavailable, they then go to the next priority queue manager.
- **CLWLUSEQ** specifies the behavior of an MQPUT operation if there is a local instance of a queue and at least one remote cluster instance. If QMGR is specified, the behavior is as specified by the CLWLUSEQ parameter of the queue manager definition. If you specify ANY, the queue manager treats the local queue as another instance of the cluster queue for the purposes of workload distribution. If you specify LOCAL, the local queue is the only target of the MQPUT operation.

- **CLWLRANK** specifies the rank of the queue for the purposes of cluster workload distribution. This parameter is valid only for local, remote, and alias queues. The value must be in the range zero through 9 where zero is the lowest rank and 9 is the highest
- **CLWLMRUC** The number of most recently chosen channels that the cluster workload management algorithm uses to restrict the number of active outbound cluster channels.
- **CLWLWGHT** applies weight to a channel, so that the proportion of messages sent down the channel can be controlled. Use this attribute to ensure that machines with more processing power are sent more messages. The higher the channel weight, the more messages are sent over that channel. The value must be in the range of 1 to 99, where 1 is the lowest rank and 99 is the highest.
- **NETPRTY** sets the priority for the network connection. If there are two possible routes to a queue manager within the cluster, the highest priority channel available is used first. Use this attribute to make one network a primary and another one the backup.

Cluster workload algorithm

1. Queue PUT(ENABLED/DISABLED)
2. Local instance (CLWLUSEQ)
3. Channel rank (CLWLRANK)
4. Queue rank (CLWLRANK)
5. Channel status
 - Inactive, running
 - Binding, initializing, starting, stopping
 - Retrying
 - Requesting, paused, stopped
6. Channel net priority (NETPRTY)
7. Channel priority (CLWLPRTY)
8. Queue priority (CLWLPRTY)
9. Most recently used (CLWLMRUC)
10. Least recently used with channel weighting (CLWLWGHT)

© Copyright IBM Corporation 2008

Figure 11-40. Cluster workload algorithm

WM203 / VM2032.0

Notes:

The workload management algorithm is used when determining the final destination for messages being put onto cluster queues. These rules are influenced by the settings applied to the attributes listed for queues, queue managers, and channels.

The checklist represents the order that is checked by the algorithm to prioritize the attributes.

Workload balancing BIND options

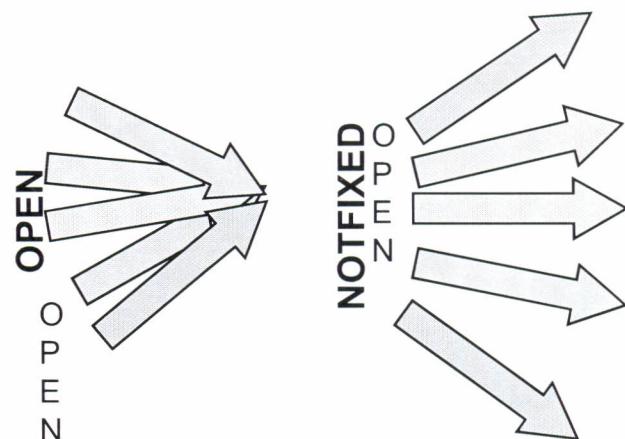
- Bind on open
 - Messages are bound to a destination chosen at MQOPEN
 - All messages are put using open handle are bound to the same destination
- Bind not fixed
 - Each message is bound to a destination at MQPUT
 - Workload balancing done on every message
 - **Recommended** – no affinities are created

Application options for MQOPEN:

MQOO_BIND_ON_OPEN
 MQOO_BIND_NOT_FIXED
 MQOO_BIND_AS_Q_DEF (default)

DEFBIND queue attribute

OPEN (default)
 NOTFIXED



© Copyright IBM Corporation 2008

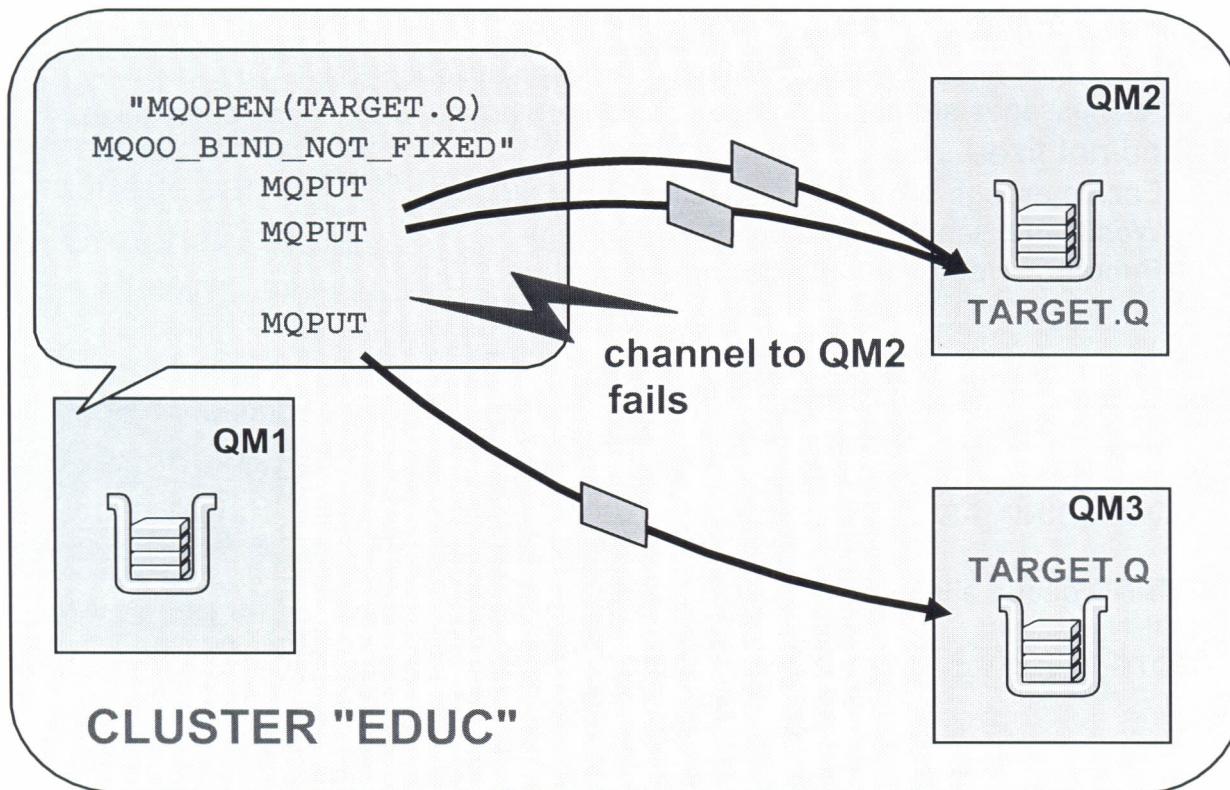
Figure 11-41. Workload balancing BIND options

WM203 / VM2032.0

Notes:

- An option on the MQOPEN call, the MQOO_BIND_ON_OPEN option, allows you to specify that, when there are multiple instances of the same queue within a cluster, the target queue manager needs to be fixed. That is, all messages put to the queue specifying the object handle returned from the MQOPEN call must be directed to the same queue manager using the same route.
- If you do not want to force all your messages to be written to the same destination, specify MQOO_BIND_NOT_FIXED on the MQOPEN call to select a destination at MQPUT time, that is, on a message-by-message basis.
- If you do not specify either MQOO_BIND_ON_OPEN or MQOO_BIND_NOT_FIXED, the default option is MQOO_BIND_AS_Q_DEF. Using MQOO_BIND_AS_Q_DEF causes the binding that is used for the queue handle to be taken from the DefBind queue attribute.

Continuous operation DEFBIND attribute



© Copyright IBM Corporation 2008

Figure 11-42. Continuous operation DEFBIND attribute

WM203 / VM2032.0

Notes:

- The cluster workload exit (built-in or user supplied) is called either when a cluster queue is opened by an MQOPEN or MQPUT1 call or when a message is put to a queue using MQPUT.
- If the target queue manager chosen at the time of an MQPUT call is unavailable, or fails while the message is still on the transmission queue, the exit is called again to select a new target queue manager.
- The queue attribute DEFBIND determines if rerouting is performed while a queue is opened:
 - DEFBIND(NOTFIXED) - Behaves as shown in the figure, with a round-robin distribution of messages to all available TARGET.Qs in the cluster.
 - DEFBIND(OPEN) - The destination queue is selected at MQOPEN time and cannot be changed until MQCLOSE.
- This attribute may be overwritten by the application using appropriate Open Options.

- Ensure that all of the instances of the queue have the same priority, default persistence, and defbind values.

Workload management topic summary

Having completed this topic, you should be able to:

- Describe how clusters assist with workload management
- List the attributes that affect the workload balancing algorithm
- Explain how application considerations play a role in workload balancing

© Copyright IBM Corporation 2008

Figure 11-43. Workload management topic summary

WM203 / VM2032.0

Notes:

Checkpoint questions

1. Which describes the main purpose of clustering?
 - a. Simplified administration
 - b. Workload balancing
 - c. Flexible connectivity
 - d. To enable publish/subscribe
 - e. All of the above
2. True or false: To send cluster messages to a partial repository QMGR from a full repository QMGR, standard sender-receiver channels are required.
3. True or false: The SYSTEM.CLUSTER.COMMAND.QUEUE holds inbound and outbound administrative messages.
4. What is the default name of the system queue which holds information about the repository?
5. True or false: Remote queue definitions are not required when using clusters.
6. What does REPOS(YES) mean on a REFRESH CLUSTER command?
7. What is the queue attribute DEFBIND used for?

© Copyright IBM Corporation 2008

Figure 11-44. Checkpoint questions

WM203 / VM2032.0

Notes:

Unit summary

Having completed this unit, you should be able to:

- Explain the purpose and use of clustering
- Define a cluster and list the components involved
- Describe the difference between a full and a partial repository
- Outline the steps required to set up a basic cluster
- Define the queue manager attributes that are related to the cluster
- List queue and channel definitions
- List the commands used to control clusters
- Describe the administration tasks that must be considered in a clustered environment
- Describe how clusters assist with workload management
- List the attributes that affect the workload balancing algorithm
- Explain how application considerations play a role in workload balancing

© Copyright IBM Corporation 2008

Figure 11-45. Unit summary

WM203 / VM2032.0

Notes: