

**Chapter
2**

**SHARED QUEUES
AND QUEUE-SHARING
GROUPS**

*Get on the
Fast Track!*



**SYS-ED/
Computer
Education
Techniques, Inc.**

Objectives

You will learn:

- Purpose and features of shared queues.
- Where the definition for a shared queue is stored.
- Storing messages in a coupling facility.
- Use of channels in queues.



1 Shared Queues

A shared queue is a type of local queue. The messages on that queue can be accessed by one or more queue managers that are in a sysplex.

The queue managers that can access the same set of shared queues form a group called a queue-sharing group. Messages can be accessed by any queue manager. A shared queue can be accessed by any queue manager in the queue-sharing group.

A message can be put on a shared queue on one queue manager, and the same message can then be received from a different queue manager. This will serve to provide a rapid mechanism for communication within a queue-sharing group that does not require channels to be active between queue managers.

The messages on a shared queue are stored in the zSeries Coupling Facility.

An application can connect to any of the queue managers within the queue-sharing group. Since all the queue managers in the queue-sharing group can access all of the shared queues, the application does not depend on the availability of a specific queue manager. Any queue manager in the queue-sharing group can service the queue.

1.1 Queue Definition – Shared by All Queue Managers

The definition of a shared queue is stored in a DB2 shared database called the shared repository.

This definition means that queues need only be defined once and then they can be accessed by all the queue managers in the queue-sharing group. This will result in fewer definitions having to be made.

By contrast, the definition of a non-shared queue is stored on page set zero of the queue manager that owns the queue. A shared queue name cannot be found if a queue with that name has already been defined on the page sets of the defining queue manager.

Likewise, a local version of a queue on the queue manager page sets cannot be define when a shared queue with the same name already exists.

2 Queue-sharing Group

The group of queue managers that can access the same shared queues is called a queue-sharing group. Each member of the queue-sharing group has access to the same set of shared queues.

Characteristics of Queue-sharing groups includes:

- Queue-sharing groups have a name of up to four characters.
The name must be unique in a network, and must be different from any queue manager names.
- Each queue manager has a channel initiator and its own local page sets and log data sets.
- Each member of the queue-sharing group must also connect to a DB2 system.
The DB2 systems must all be in the same DB2 data-sharing group so that the queue managers can access the DB2 shared repository used to hold shared object definitions.

These definitions, which can be of any type of WebSphere MQ object, are defined once only and can then be used by any queue manager in the group. These are called global definitions.

A specific data-sharing group can be referenced by more than one queue-sharing group. The name of the DB2 subsystem and the data-sharing group used by a queue manager is specified in the WebSphere MQ system parameters at startup.

When a queue manager has joined a queue-sharing group, it has access to the shared objects defined for that group, and can be used to define new shared objects within the group. If shared queues are defined within the group, this queue manager can be used to put messages to and get messages from those shared queues.

Messages held on a shared queue can be retrieved by any queue manager in the group.

An MQSC command can be entered once and it then can be executed on all queue managers within the queue-sharing group as if it had been entered at each queue manager individually.

3 Shared Queue Messages – Where they are Held?

The messages in shared queues are stored on list structures in the zSeries Coupling Facility. They can be accessed by many queue managers in the same sysplex.

All queue managers also maintain their own logs and page sets to use non-shared local queues, and store definitions of private objects on page set zero. Messages that are put on to shared queues are not stored on page sets.

MQPUT and MQGET operations on persistent messages, on both shared and non-shared queues, are recorded on the queue manager log. This will serve to minimize the risk of data loss in the event of a Coupling Facility or page set failure.

4 The Coupling Facility

The messages held on shared queues are stored inside a Coupling Facility. The Coupling Facility resides outside any of the z/OS images in the sysplex and is typically configured to run on a different power supply. The Coupling Facility will be resilient to software failures and can be configured so that it is resilient to hardware failures or power-outages. Messages stored in the Coupling Facility will be highly available.

WebSphere MQ uses Coupling Facility list structures to store messages. The maximum length for messages on a shared queue is 63 KB. Each Coupling Facility list structure used by WebSphere MQ is dedicated to a specific queue-sharing group; a Coupling Facility can hold structures for more than one queue-sharing group.

Queue managers in different queue-sharing groups cannot share data.

- Up to 32 queue managers in a queue-sharing group can connect to a Coupling Facility list structure at the same time.
- A single Coupling Facility list structure can contain up to 512 shared queues.

The amount of message data is limited by the size of the list structure.

5 Shared Inbound Channels

Each channel initiator in the queue-sharing group starts an additional listener task to listen on a generic port. This generic port is made available to the network through one of the technologies mentioned above. This means that an inbound network attach request for the generic port can be dispatched to any one of the listeners in the queue-sharing group that are listening on the generic port.

A channel can only be started on the channel initiator to which the inbound attach is directed if the channel initiator has access to a channel definition for a channel with that name.

A channel definition can be defined to be private to a queue manager or stored on the shared repository and available anywhere (a global definition). This means that a channel definition can be made available on any channel initiator in the queue-sharing group by defining it as a global definition.

There is an additional difference when starting a channel through the generic port; channel synchronization is with the queue-sharing group and not with an individual queue manager. For example, consider a client starting a channel through the generic port.

When the channel first starts, it might start on queue manager QM1 and messages flow. If the channel stops and is restarted on queue manager QM2, information about the number of messages that have flowed is still correct because the synchronization is with the queue-sharing group.

6 Shared Outbound Channels

An outbound channel is considered to be a shared channel if it is taking messages from a shared transmission queue. If it is shared, it holds synchronization information at queue-sharing group level.

This means that the channel can be restarted on a different queue manager and channel initiator instance within the queue-sharing group if the communications subsystem, channel initiator, or queue manager fails. Restarting failed channels in this way is a feature of shared channels called peer channel recovery.

Workload balancing is an important feature. An outbound shared channel is eligible for starting on any channel initiator within the queue-sharing group, provided that the channel has not been specified to be started on a particular channel initiator.

7 Shared Channel Status

The channel initiators in a queue-sharing group maintain a shared channel-status table in DB2. This records which channels are active on which channel initiators.

The shared channel-status table is used when there is a channel initiator or communications system failure. It indicates which channels need to be restarted on a different channel initiator in the queue-sharing group.

8 Queues and Queuing

Intra-group Queuing

A fast message transfer can be performed between queue managers in a queue-sharing group without defining channels. This uses a system queue called the `SYSTEM.QSG.TRANSMIT.QUEUE`, which is a shared transmission queue.

Each queue manager in the queue-sharing group starts a task called the intra-group queuing agent, which waits for messages to arrive on this queue that are destined for their queue manager. When such a message is detected, it is removed from the queue and placed on the correct destination queue.

Standard name resolution rules are used but, if intra-group queuing is enabled and the target queue manager is within the queue-sharing group, the `SYSTEM.QSG.TRANSMIT.QUEUE` is used to transfer the message to the correct destination queue manager instead of using a transmission queue and channel. It can be used only to move messages with a message length of up to 63 KB, including the transmission-queue header. For shared queues, 63 KB is the maximum message length.

Initiation Queues

When defining a shared initiation queue, it will not be necessary to have a trigger monitor running on every queue manager in the queue-sharing group. The only prerequisite condition is that there must be at least one trigger monitor running.

A shared initiation queue can also be used. It can be used even if there is a trigger monitor running on each queue manager in the queue-sharing group.

If there is a shared application queue and the trigger type of `EVERY` or a trigger type of `FIRST` with a small trigger interval is used, the initiation queue should always be a shared queue.

Dead-letter Queue

Dead-letter queues should not be defined as a shared queue. This is because shared queues cannot hold messages with a size greater than 63 KB.