WebSphere MQ for z/OS
Introduction to Shared Queues

Agenda

- What shared queues are
  - Shared queues
  - Queue-sharing groups
  - Coupling Facility (CF) structures
  - Persistence and transaction integrity
- Configuring channels with shared queues
  - Inbound channel configurations
  - Outbound channel configurations
- Exploiting shared queues
  - Availability benefits of queue sharing
  - Scalability
- New features in V6.0
**Shared Queues?**

- **Function:**
  - Multiple Queue Managers can access the same shared queue messages
  - Multiple Queue Managers can access the same shared queue objects

- **Benefits**
  - Availability for new messages
  - Availability for old messages
  - Pull workload balancing
  - Scalable capacity
  - Low cost messaging within a sysplex

---

**Mover**

- Application
- Mover (CHIN)
- Queue manager
- Xmit queue

- Mover (CHIN)
- Application
- Queue manager
- Target queue
Introduction to Shared Queues

**Shared Queues**

- **Application**
- **QMgr**
- **Coupling Facility (CF)**
- **Shared Queue**

- **Same sysplex**
- **Physical message size**
  - < 63KB (V5.3.1)
  - < 100MB (V6)

**Intra-Group Queue**

- **Application**
- **Queue Manager**
- **Target Queue**
- **SYSTEM.QSG.TRANSMIT.QUEUE**
- **Intra-group queue**
Creating a queue-sharing group

Use the CSQ5PQSG utility to create a QSG:

1. Add the QSG into the DB2 tables:
   ```
   //stepname EXEC PGM=CSQ5PQSG,
   //   PARM='ADD QSG, qsg-name, dsg-name, DB2-ssid'
   ```

2. Add the queue managers into the DB2 tables as members of the QSG:
   ```
   //stepname EXEC PGM=CSQ5PQSG,
   //   PARM='ADD QMGR, qmgr-name, qsg-name, dsg-name, DB2-ssid'
   ```

<table>
<thead>
<tr>
<th>qsg-name</th>
<th>Name for the queue-sharing group</th>
</tr>
</thead>
<tbody>
<tr>
<td>qmgr-name</td>
<td>Name of the queue manager</td>
</tr>
<tr>
<td>dsg-name</td>
<td>Name of the DB2 data-sharing group</td>
</tr>
<tr>
<td>DB2-ssid</td>
<td>DB2 subsystem ID</td>
</tr>
</tbody>
</table>
Creating CF structures and shared queues

- Define a structure to z/OS (not to WebSphere MQ) by updating the CFRM policy (see System Setup Guide):
  - Structure is known to WebSphere MQ by its 12-character str-name.
  - Structure is known to z/OS by the 16-character name formed by:
    - qag-name | str-name (Application structures)
    - qag-name || CSQ_ADMIN (Administration structure)

- Define a shared queue using the DEFINE QLOCAL command on any queue manager in the QSG:
  - DEFINE QLOCAL(queue-name) QSGDISP(SHARED) CFSTRUCT(str-name)

- z/OS creates the structure when required (first use).
- WebSphere MQ creates the queue when required (first use).
Safeguarding against CF failure

- Administration structure updates are logged so that this structure can be restored.
- Coupling Facilities are very rugged (zSeries processor technology).
- CF can have its own separate power supply.
- CF can have nonvolatile memory (battery power backup).
- Lost application structures can be restored from backups and logs.
Introduction to Shared Queues

WebSphere Software

Transaction integrity

Original state of shared queue

Message
Message
Message
Message

Transaction in flight

Message
Message
Message
Message

Queue manager writes the UOWD to the administration structure

UOWD on the administration structure -- allows another queue manager to complete the transaction

Queue manager commits the transaction

Transaction complete -- UOWD deleted

Message
Message
Message
Message

Agenda

- What shared queues are
  - Shared queues
  - Queue-sharing groups
  - Coupling Facility (CF) structures
  - Persistence and transaction integrity

- Configuring channels with shared queues
  - Inbound channel configurations
  - Outbound channel configurations

- Exploiting shared queues
  - Availability benefits of queue sharing
  - Scalability

- New features in V6.0
Inbound connection to local port/LU of specific queue manager's mover

**Notes**

This chart is the first of three which show different ways to configure inbound channels.

This configuration uses the local port (TCP/IP) or logical unit (LU6.2) to connect to a specified mover (queue manager). It works almost exactly the way existing channel configurations work, including using the local sync queue for the channel — but:

Because the queue manager is part of a queue-sharing group, the channel can put messages directly onto a shared queue. That is, the target application can be on any of the queue managers in the QSG.

The chart shows other ports/LUs not used by this configuration.
This configuration uses VTAM generic resources (LU6.2) or dynamic DNS (TCP/IP) to connect to any mover in the queue-sharing group.

The chart shows that the connection has "selected" the mover shown at top right, but a subsequent connection could select another mover in the QSG.

Notice that the mover uses the shared sync queue for this channel (because access was through the shared LU or port). The shared sync queue is: SYSTEM.QSG.CHANNEL.SYNCQ.

If the channel loses its connection (for example, because this queue manager fails), it can connect to a different mover. But this different mover can resynchronize the channel using the shared sync queue.

You can configure VTAM generic resources or dynamic DNS to use the z/OS workload manager (WLM) to select the "least busy" mover -- providing load balancing.

If the target queue for a put is not shared then the same private queue must be defined on each of the queue managers in the QSG.
Inbound connection to shared port/LU of specific queue manager's mover.

**NOTES**

This chart shows that you can connect directly to the shared port of a specified mover -- that is, you do not have to use VTAM generic resources or dynamic DNS.

This allows you to use an "external" router to select which mover to connect to.

By connecting to the shared port the connection uses the shared sync queue. This allows correct resynchronization if the channel fails and reconnects to the shared port of a different mover.

A disadvantage of an external router is that it does not use the workload manager to identify the "best" (least busy) mover.
Outbound channel configurations

This chart shows the two possible configurations for outbound channels.

At the top is a private (or local) outbound channel. It works exactly the way existing channel configurations work:
- Private transmission queue local to the mover
- Private synchronization queue local to the mover.

Below is a shared outbound channel:
- Shared transmission queue -- any application in the QSG can use it
- Shared synchronization queue.

A shared outbound channel can start on any mover -- WebSphere MQ selects the "best" (least busy) mover.

If a shared outbound channel fails (communication, mover, or queue manager failure), the channel can restart automatically on another mover. This is called peer channel recovery.

Shared queue restrictions apply to shared transmission queues, for example:
- Maximum message length is 63KB if version is < V6
- CF capacity is limited (compared to DASD).
Introduction to Shared Queues

WebSphere Software

Agenda

- What shared queues are
  - Shared queues
  - Queue-sharing groups
  - Coupling Facility (CF) structures
  - Persistence and transaction integrity
- Configuring channels with shared queues
  - Inbound channel configurations
  - Outbound channel configurations
- Exploiting shared queues
  - Availability benefits of queue sharing
  - Scalability
- New features in V6.0

WebSphere Software

Availability – two servers

Unplanned outages

Planned outages

- Full availability
- Reduced capacity (off-peak)
- Reduced capacity
- Outage
- Do not do this
## Availability – example

<table>
<thead>
<tr>
<th></th>
<th>One server</th>
<th>Two servers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planned outages</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Unplanned outages</td>
<td>0.5</td>
<td>0.002</td>
</tr>
<tr>
<td>Total</td>
<td>1.5</td>
<td>0.002</td>
</tr>
</tbody>
</table>

Outages in hours per month

## Shared queue scaling – non persistent

![Maximum message rate -- single queue](chart.png)

- Local: 4100 messages per second
- 1 QMG: 5850 messages per second
- 2 QMGs: 10900 messages per second
- 3 QMGs: 16550 messages per second
Introduction to Shared Queues

Agenda

- What shared queues are
  - Shared queues
  - Queue-sharing groups
  - Coupling Facility (CF) structures
  - Persistence and transaction integrity
- Configuring channels with shared queues
  - Inbound channel configurations
  - Outbound channel configurations
- Exploiting shared queues
  - Availability benefits of queue sharing
  - Scalability
- New features in V6.0

New features in V6.0

- Large shared queue messages
- Admin Structure Failure Tolerance
- IGQ Double-hop avoidance
- Multiple CICS bridge monitors on a single queue
Shared queue benefits

- No mover between servers in the QSG.
- Pull load-balancing for servers.
- Availability from multiple servers.
- Workload-balancing for movers.
- Availability from shared channels.
- Simplified configuration management from shared object definitions and command scoping.
- Flexible capacity management.