Using the MQ 7 Verbs in CICS programs
Session 09669

Lyn Elkins – elkinsc@us.ibm.com

Agenda

Some of the “new” functionality that will be discussed:

1) Publishing
2) Subscribing
3) Cooperative Browsing
4) Message properties
5) Call back
6) Flagrant Promotion of sessions and samples
Publishing with MQ and CICS

• To publish in MQ is to:
  • Put a message to a TOPIC, not a queue
    • There is no new verb (No MQPUB)
  • The publishing application can be blissfully ignorant
    • Are there any subscribers?
    • How many subscribers are there?
    • Where are the subscribers?
  • Publishing is easy

Publishing with MQ and CICS - Notes

• To publish in MQ is to:
  • Put a message to a TOPIC, not a queue
    • There is no new verb (No MQPUB)
    • You can use a QALIAS to change and existing MQPUT to point to a topic rather than a queue
  • The publishing application can be blissfully ignorant
    • But they do not have to be
      • You can get a warning (MQPMO-WARN-IF-NO-SUBS-MATCHED) returned on the MQPUT
  • Publishing may be too easy, and if the message is retained it can live on (like a drunk text message sent to your mother/spouse/significant other)
Publishing with MQ and CICS - Warnings

- Publishing messages can be an efficient way to get the same messages set to multiple interested parties, but:
  - If there is only 1 subscriber, it is more costly than point to point
  - Publications may need expiration more than ‘normal’ messages
  - Retained publications can clutter up the queue
  - Be careful when considering publication of sensitive information

Publishing with MQ and CICS – Warnings - Notes

- Publishing messages can be an efficient way to get the same messages set to multiple interested parties, but:
  - If there is only 1 subscriber, it is more costly than point to point
    - Please see SupportPac MP1F for information on the costs of pub/sub on z/OS
  - Publications may need expiration more than ‘normal’ messages
    - Durable subscriptions queues can fill
    - Subscriptions, even non-durable remain active until a transaction ends to the subscription is stopped
      - *Publications will continue to be delivered after the application has closed the subscription queue*
Publishing with MQ and CICS – Coding notes

- Published messages are typical MQ messages:
  - Persistent or not
  - Expire
  - Supply reply to information
  - Use report options

Subscribing with MQ and CICS

- Publishing is easy, subscribing is a little more challenging!
- MQSUB is the new verb
  - You can use administered subscriptions to do a quick enablement.
  - It registers the interest in the topic with the queue manager
  - Tells MQ where the message should be delivered
  - Defines the subscription duration
- IT DOES NOT RETURN ANY MESSAGES TO THE PROGRAM!!
  - The application has to open the queue and MQGET the messages
Subscribing with MQ and CICS - Warnings

- Using ‘managed’ subscriptions on z/OS should be carefully evaluated
  - The cost of temporary dynamic queues is much higher than using a predefined queue
- To repeat:
  - MQSUB DOES NOT RETURN ANY MESSAGES TO THE PROGRAM!!
    - The application has to open the queue and MQGET the messages

Subscribing with MQ and CICS

- Subscription duration considerations
  - Even a non-durable subscription may leave messages
    - A new kind of orphaned message
  - To prevent this:
    - Publications can all be set to expire
    - Programming model can include reading to the end of the queue AFTER the subscription is closed
    - Periodic clean up
    - Use Temporary dynamic queues
- Strong suggestion - give subscriptions a name
Subscribing with MQ and CICS - Notes

- Subscription duration
  - These publications are a new form of orphaned message
    - The typical programming model for CICS with MQ is
      - Open queue
      - Get in loop until you get a 2033, or a threshold is met
      - Close queue
      - Return control to CICS
  - The window if opportunity for these orphans is between the close of the queue and the end of the transaction.
  - Using temporary dynamic queues on z/OS is expensive, and not seriously recommended
- Subscription names:
  - Develop naming standards for subscriptions, just like other resources
  - Giving a subscription a recognizable name makes it easier to identify

Subscribing with MQ and CICS

- Multiple subscribers can use the same target queue
  - Each subscriber supplies a unique correlation ID
  - The MQGETs then match on the correlation to retrieve 'their' messages
- The target queue is just a queue
  - Nothing identifies it as strictly a subscription destination
  - Other messages may be put on the queue
  - The MQGET, if not selective, will get the next message available
Subscribing with MQ and CICS - Notes

- Using a common target queue
  - Multiple application or clients can use the same queue – saving queue definitions, and other resources
  - Remember – there is nothing special about the destination queue used by a subscription

Cooperative Browsing

- Designed for work dispatching
  - Multiple applications can browse and dispatch work, without competing for messages
  - Can reduce the costly transaction starts for messages already processed by other instances
  - Allows new (or newly committed) messages to be 'in stream'
  - Uses a 16-byte message token to uniquely identify the message
  - You must alter existing dispatching programs to take advantage of this
Cooperative Browsing

- Programming Warnings
  - Set MQGMO-VERSION to MQGMO-VERSION-4
  - Use MQGMO-BROWSE-CO-OP if sequence is not important
    - Sum of
      - MQGMO_BROWSE_FIRST,
      - MQGMO_UNMARKED_BROWSE_MSG, and
      - MQGMO_MARK_BROWSE_CO_OP
  - Use MQGMO_BROWSE_HANDLE if sequence is important
    - Sum of
      - MQGMO_BROWSE_FIRST,
      - MQGMO_UNMARKED_BROWSE_MSG, and
      - MQGMO_MARK_BROWSE_HANDLE

Cooperative Browsing - Notes

- Warning!
  - Make sure you have current maintenance
  - Notice the ‘BROWSE FIRST’, this is used even when you are going thru the entire queue
    - This is how new high priority messages and newly committed messages are picked up
  - MQGMO-BROWSE-CO-OP = 18874384 or x’01200010’
  - MQGMO_BROWSE_HANDLE = 17825808 or x’01100010’
Message Properties – ‘the side data’

- Message properties are information about the message not part of the message body
- MQMD and MQMDE fields are also properties
  - They do not require the new verbs
- Suggested uses:
  - Message selection:
    - Red messages processed by transaction ABCD
    - Green messages processed by QERF
  - Adding processing data or status
    - ‘I have been thru transaction ABCD’
- Formerly RFH2 data

Message Properties – ‘the side data’ - Notes

- Formerly RFH2 data
- There are a number of places where options have been added to provide ‘old style’ delivery of the properties:
  - On queue definitions:
    - PROPCTL Values
      - COMPAT – the default value. If the message contains a property with a prefix of mcd., jms., usr., or mqext., all message properties are delivered to the application in an MQRFH2 header. Otherwise all properties of the message, except those contained in the message descriptor (or extension), are discarded and are no longer accessible to the application. Applications which expect JMS related properties to be in an MQRFH2 header in the message data to continue to work unmodified. If a message handle is supplied then the behavior is to return the properties in the message handle. All properties that are identified with a special type are delivered in the RFH2 headers, others are discarded
      - ALL – message properties are delivered as RFH2 headers
      - FORCE - Properties are always returned in the message data in an MQRFH2
      - NONE – All properties are removed
  - On the MQGET:
    - MQGMO-PROPERTIES-FORCE-MQRFH2
    - MQGMO-PROPERTIES-COMPATIBILITY
Message Properties – ‘the side data’

- Message properties are associated with a message via a Message Handle
- Multiple message properties associated with the handle
- New verbs have been added to create and delete the handles
- New verbs have been created to add, inquire and delete message properties

Message Properties – Warnings

- Performance
  - No Indexing
    - Deep queues may, no will, see higher CPU and response time when selecting by a property
  - The message size can increase dramatically when properties are added
    - Potential performance impact
    - Potential resource (bufferpool, pageset, logs) impact
  - ‘My properties are gone’
    - Check your queue properties
Message Properties – The New Verbs & Options

- Verbs:
  - MQCRTMH – to create the message handle
    - Note the message handle is defined as S9(18) BINARY
  - MQDLTMH – to delete the message handle
  - MQSETMP – Set message property
    - Will initially set or update a property value
  - MQINQMP - Inquire message property
    - Will return the value of a message property
  - MQDLTMP - Delete message property

Message Properties – The New Verbs & Options

- Some of the New Options, Fields, and Value:
  - On the PMO:
    - PMO version needs to be ‘3’
    - MQPMO-ORIGINALMSGHANDLE
    - MQPMO-NEWMSGHANDLE
    - MQPMO-ACTION
  - On the GMO:
    - GMO version needs to be ‘4’
    - MQGMO-MSGHANDLE
Message Properties – Coding Observations

- Coding Observations
  - Property names are more restrictive than documented
    - No spaces
    - Cannot start with a number
    - And there may be others
  - When using the Message handle and properties API:
    - The HCONN value may change
      - This is normal, these API calls are handled in the interface, not by the queue manager
      - Under CICS you do not have to save and restore this value

Adding message properties example

- Simple program that:
  - Reads messages from a queue
  - Add two properties to the message
  - Writes the message to an outbound queue
The inbound messages

- Using the MQ Explorer to put two messages on the inbound queue
- Note the Data length field for both

The inbound messages - continued

- The properties of both inbound messages looked as shown
The outbound messages

- The processing program added two message properties to each message
- The message body remained the same
- Note the new ‘Data length’ values

The properties of both outbound messages look as shown

The panel shows ‘Named Properties’ are attached to the message
The outbound messages - continued

- Clicking on that tab will display the added properties

Async Message Consuming

- Aka MQCB and MQCTL verbs
  - Session 09602 covered in detail
  - Registers interest in specific messages and ‘calls back’ to the defined routine
    - The application can continue with non-MQ work
    - The called routine gets the message in a buffer
    - No ‘Get Wait’
    - No ‘Triggering’
Async Message Consumer

- More expensive than an MQGET with WAIT when only 1 queue is being used
  - Can be less expensive when more queues are being used
  - See MP1F
- Sample programs:
  - CSQ4CVRG - Registration Client
  - CSQ4CVEV - Event handler CSQ4CVEV
  - CSQ4CVCN - Simple Message Consumer
  - CSQ4CVCT - Control Message Consumer
  - CSQ4CVPT - Messaging Client
- Description may be found at:

Flagrant Promotion

- Other SHARE sessions:
  - MQ Pub/sub:
    - MQ Publish/Subscribe for additional details - session 09419
    - CICS TS MQ Update – session 09602
  - Other verbs:
    - The MQ API for dummies - the basics
    - So, what else can I do? - MQ API beyond the basics
- ATS Samples – COBOL CICS ‘one function’ samples
  - QPUB – Triggered publication program, with resource definitions and test sample
    - http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/PRS4549
  - QSUB – Triggered subscription program - in trial now, we’ll be glad to email you a copy!
  - QBRS – In development - Cooperative browse sample
  - QPR1 – In development – Programs to set ‘where I’ve been’ message properties
The rest of the week ……

<table>
<thead>
<tr>
<th>Time</th>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
</tr>
</thead>
<tbody>
<tr>
<td>08:00</td>
<td></td>
<td>More than a buzzword: Extending</td>
<td>Batch, local, remote, and traditional MVS - file</td>
<td>Lyn's Story Time - Avoiding the MQ</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>the reach of your MQ messaging with Web 2.0</td>
<td>processing in Message Broker</td>
<td>Problems: Others have Hit!</td>
<td></td>
</tr>
<tr>
<td>09:30</td>
<td>WebSphere MQ 101: Introduction to the</td>
<td>The Do's and Don'ts of Message Broker</td>
<td>So, what else can I do? - MQ API beyond the</td>
<td>MQ Project Planning Session</td>
<td></td>
</tr>
<tr>
<td></td>
<td>world's leading messaging provider</td>
<td>Performance</td>
<td>basics</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:00</td>
<td>MQ Publish/Subscribe</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12:15</td>
<td>MQ Freesies! Top 5 SupportPlaza</td>
<td>The doctor is in. Hands-on lab and lots of</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>help with the MQ family</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01:30</td>
<td>Diagnosing problems for MQ</td>
<td>WebSphere Message Broker 101: The Swiss</td>
<td>The Dark Side of Monitoring MQ - SMF</td>
<td>Getting your MQ JMS applications running,</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>army knife for application integration</td>
<td>115 and 116 record reading and interpretation</td>
<td>with or without WAS</td>
<td></td>
</tr>
<tr>
<td>03:00</td>
<td>Keeping your eye on it all - Queue</td>
<td>The MQ API for dummies - the basics</td>
<td>Under the hood of Message Broker on z/OS - WLM,</td>
<td>Message Broker Patterns - Generate applications in an instant</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Manager Monitoring &amp; Auditing</td>
<td></td>
<td>SMF and more</td>
<td></td>
<td></td>
</tr>
<tr>
<td>04:30</td>
<td>Message Broker administration for dummies</td>
<td>All About WebSphere MQ File Transfer</td>
<td>For your eyes only - WebSphere MQ Advanced</td>
<td>Keeping your MQ service up and running -</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Edition</td>
<td>Message Security</td>
<td>Queue Manager clustering</td>
<td></td>
</tr>
<tr>
<td>06:00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Free MQ - MQ Clients and what you can do</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>with them</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Session 09669

- If you have questions, please feel free to ask!