IMQ09 - IBM MQ V9 for z/OS Wildfire Workshop

L05 – Using CAPEXPRY on z/OS

Version V6.0

October 2018

Table of Contents

Table of Contents 1
History and Tradition 3
Lab Objectives 3
General Lab Information and Guidelines 3
Step 1 – Defining and priming the queues 4
Step 2 – Enabling CAPEXPRY on the queue manager 7
Step 3 – Setting the CAPEXPRY on the queue 8
Step 4 – Test the new value - Putting new messages on the queue 11
Step 5 – Stop and Start the queue manager 12

History and Tradition

Setting message expiration has always been the prerogative of the application program and programmer. This has been decried by generations of systems administrators, because of an application tendency to not set message expiration (expiry = unlimited) or to set it to an unreasonably high value. This can mean that messages destined for a specific application can remain on a queue forever, taking up valuable space and potentially causing elongated recovery times if the messages are persistent.

***Please note that this is a departure from the normal MQ way of doing things – where the application has the final word on the message disposition*.**

Lab Objectives

This lab has the following objectives:

- To allow users to implement and test the CAPEXPRY feature as it is implemented in z/OS.

General Lab Information and Guidelines

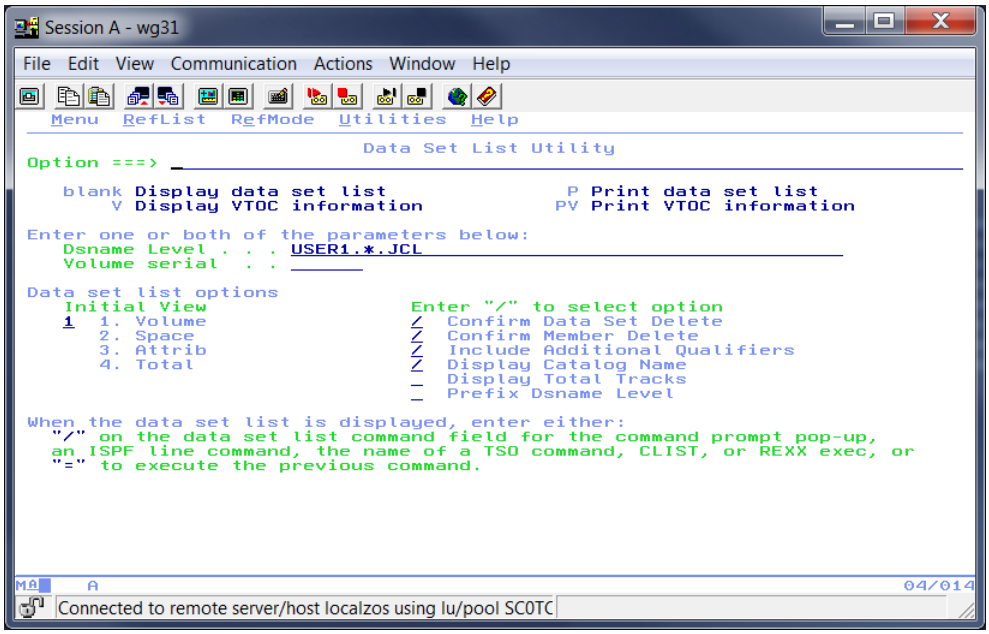
- Information required to complete this exercise will be provided on a 'worksheet' prior to the start of this exercise. Refer to this worksheet for which user identity and password are to be used and for other values, for example:
  - ✓ This exercise requires using TSO user *USER1* on the wg31.washington.ibm.com system
  - ✓ As a reminder, when a value from your worksheet should be used, the values in the instructions will be in red rather than black.
  - ✓ ***Bold italicized*** text indicates values that need to be entered on a screen.
  - ✓ *Italicized* text indicates values that are constants or names that appear on a screen.
  - ✓ **Bold** text indicates the name of buttons or keyboard keys that need to be pressed.

Step 1 – Defining and priming the queues

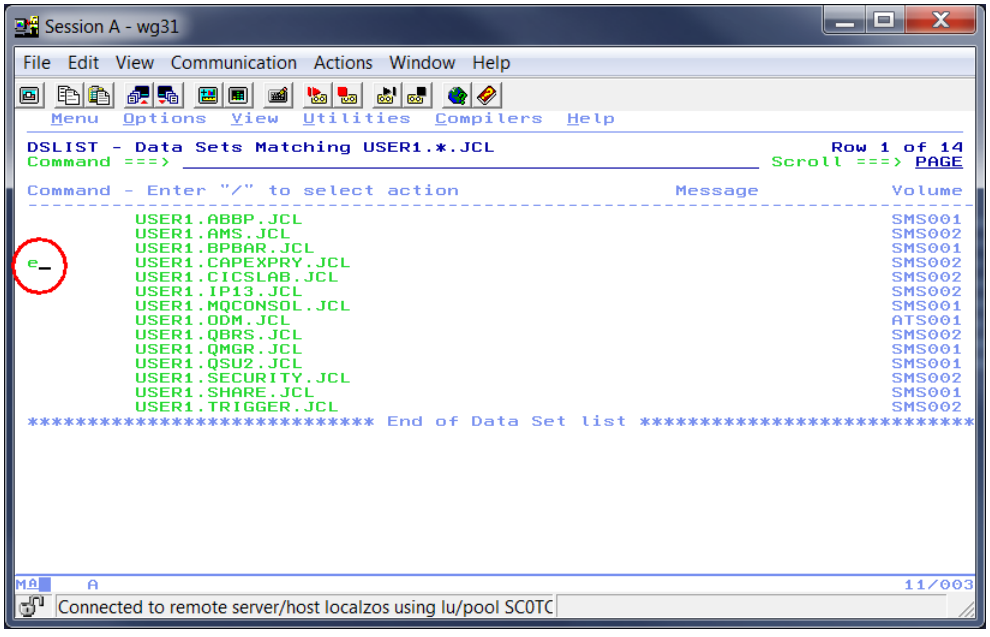
Use the MQ utility CSQUTIL job to define your queues for this lab.

____1. Logon to TSO and use the ISPF command ***=3.4*** to navigate to the dataset selection panel.

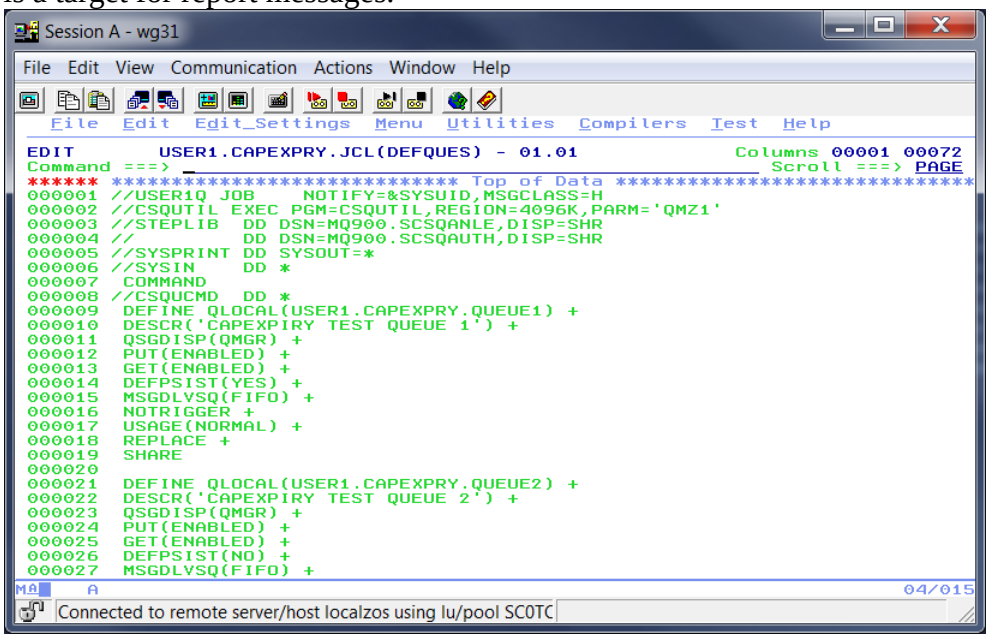
____2. Enter ***USER1.*.JCL*** in the *Dsname Level* field as shown below.



____3. From the list of JCL PDS libraries, edit the *USER1.CAPEXPRY.JCL* data set as shown below.

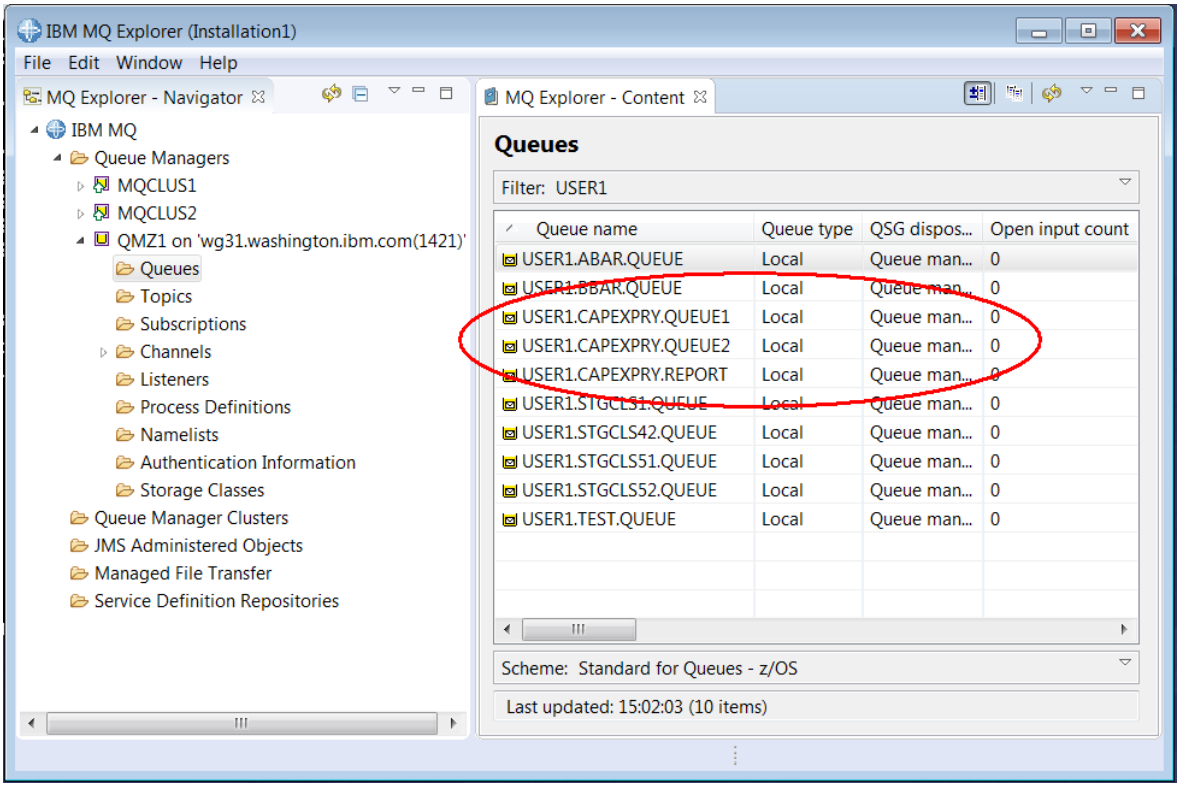


____4. Select the member *DEFQUES*. This job creates three queues that will be used in this test. The first two have the same characteristics, with one difference – the default persistence. The third queue is a target for report messages.

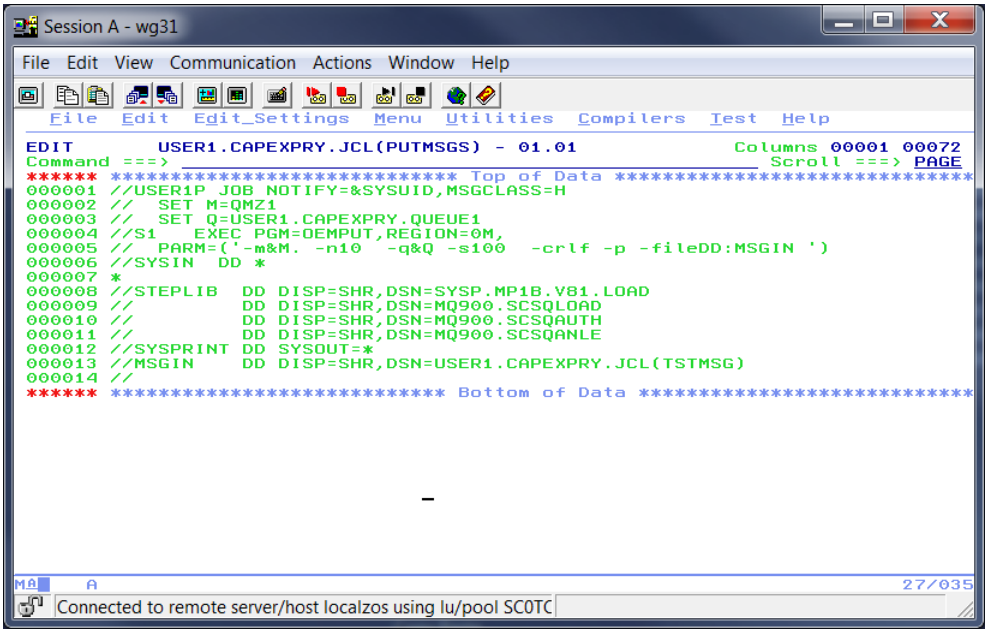


____5. Submit the JCL to create the queues and make sure the job gets a zero return code.

____6. Verify the queues were created by using the MQExplorer:



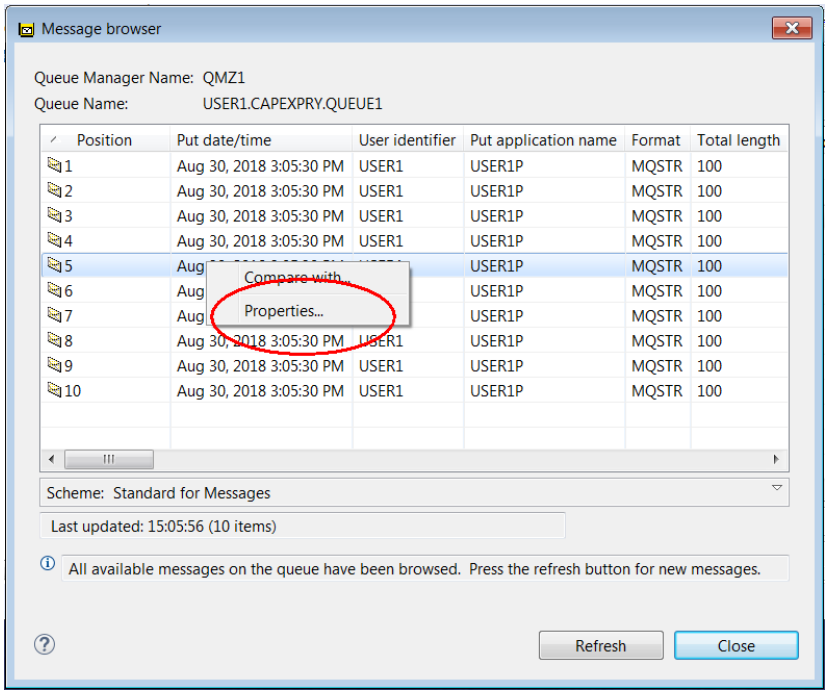
____7. Return to the member list of the *USER1.CAPEXPRY.JCL* and select member *PUTMSGS.*



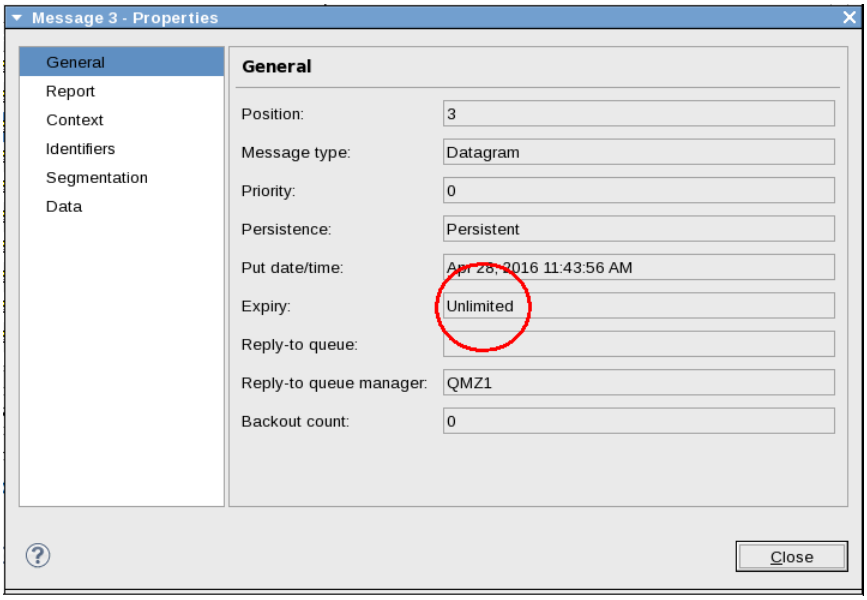
____8. This job will put 10 messages of 100 bytes on the queue specified by the *Q* parameter. In this case it is *USER1.CAPEXPRY.QUEUE1*. Submit the job and make sure it runs with a zero return code.

____9. Verify that messages were put to the queue, by using the MQ Explorer by browsing the messages on queue *USER1.CAPEXPRY.QUEUE1*. (Hint: select the queue and right mouse button click and select the *Browse Messages* option).

____10. Select one of the messages (it does not matter which one) and right mouse button click.



____11. Select *Properties* (see above) to view the message expiration value, which should be *Unlimited*. These 10 messages will never expire.



Step 2 – Enabling CAPEXPRY on the queue manager

Enabling CAPEXPRY for a queue manager requires using the MQ *RECOVER* command.

____1. Use the ISPF command ***=D.LOG*** to go to the SDSF log panel and enter the command below:

QMZ1 RECOVER QMGR (TUNE CAPEXPRY ON)

> **Tech-Tip:** *QMZ1* is the *command prefix* string of this queue manager. A command prefix (cpf) string is used to direct commands to a queue manager. On this system the queue manager's *cpf* string is the same as the queue manager's name.
>
> The leading slash is required when entering MVS commands at the SDSF prompt.

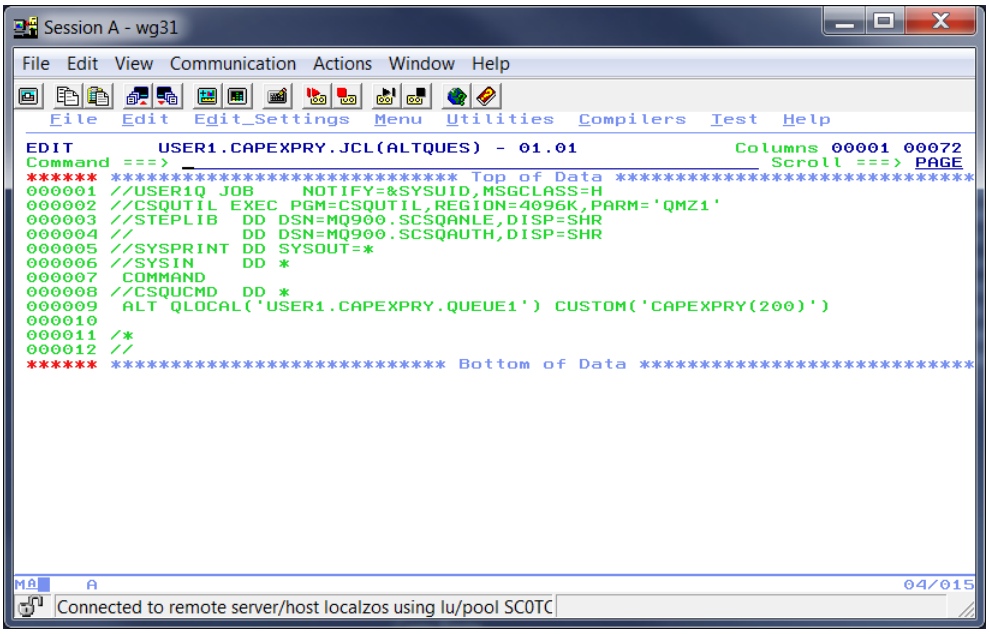
____2. You should see the messages below in the system log.

CSQI834I QMZ1 PARM CAPEXPRY = ON (Default OFF SET)
CSQ9022I QMZ1 CSQIRECP ' RECOVER QMGR' NORMAL COMPLETION

Step 3 – Setting the CAPEXPRY on the queue

Currently enabling CAPEXPRY for a queue requires the use of the queue's *CUSTOM* attribute.

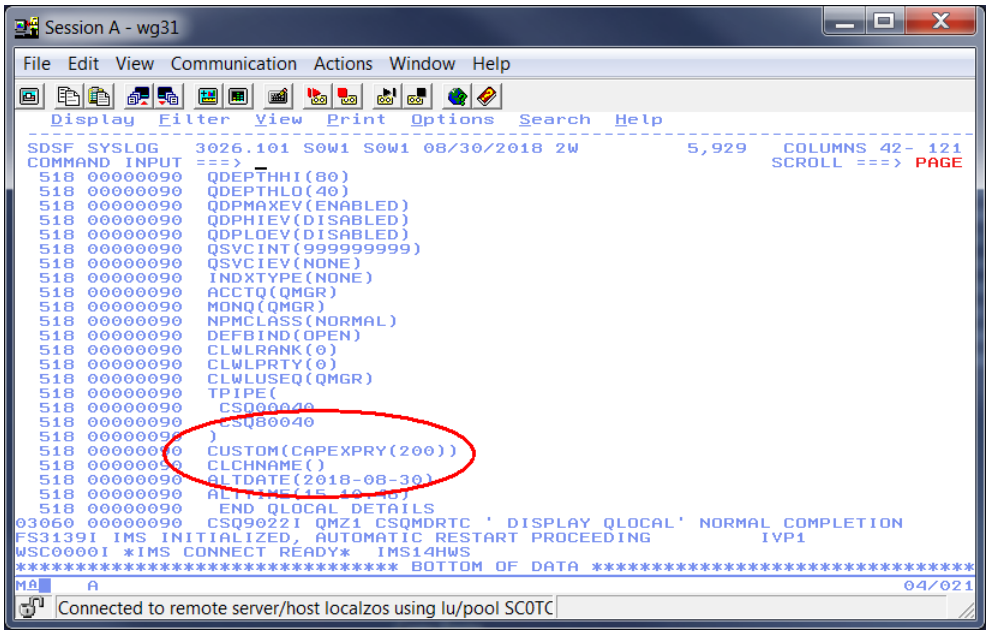
____1. Return to the *USER1.CAPEXPRY*.JCL member list and select member *ALTQUES*. The *ALTER* command in this job updates the expiration for each message put to the queue to 20 seconds, the expiration value is in tenths of seconds. Submit the job and makes sure there is a zero return code.



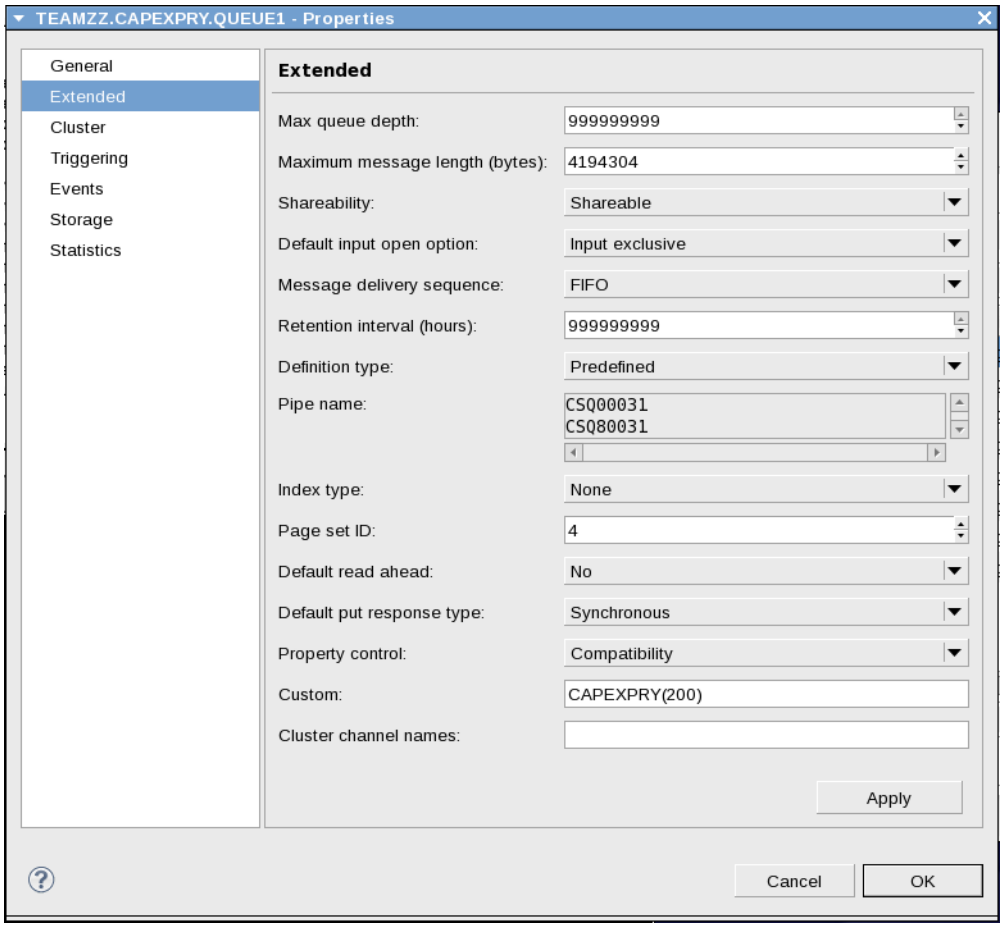
____3. Use the ISPF command ***=D.LOG*** to go to the SDSF log panel and enter the MQ command below:

QMZ1 DISPLAY QL (USER1.CAPEXPRY.QUEUE1) ALL

____2. The *CUSTOM* attribute of the queue should show the expiration details.



The value should also be shown on the *Extended* tab of the queue properties from the MQ Explorer.

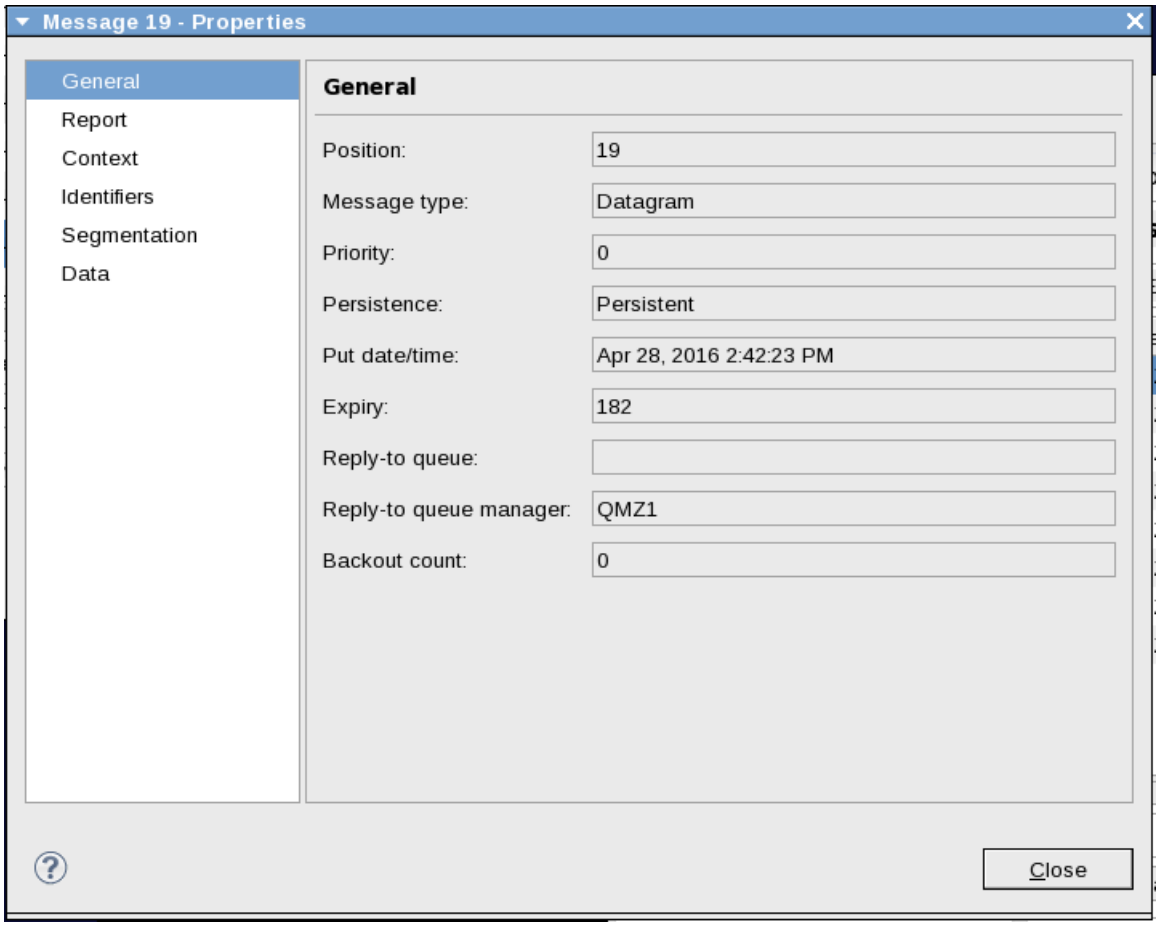


Step 4 – Test the new value - Putting new messages on the queue

____1. Return to the JCL library and select member *PUTMSGS* and submit the job for execution.

____2. When the job completes, browse the new messages on the queue. Note that you may need to refresh the browse list to display the new messages.

Select one of the new messages and display the properties. They should look something like what is shown, but the *Expiry* value may be different depending on how quickly you get to the messages.



____3. Wait a minute or so, to allow the message to expire. Re-browse the queue; the new messages should have disappeared while the original 10 messages should remain.

Step 5 – Stop and Start the queue manager

____1. From the SDSF DA ***(=D.DA***) panel enter the stop queue manager command.

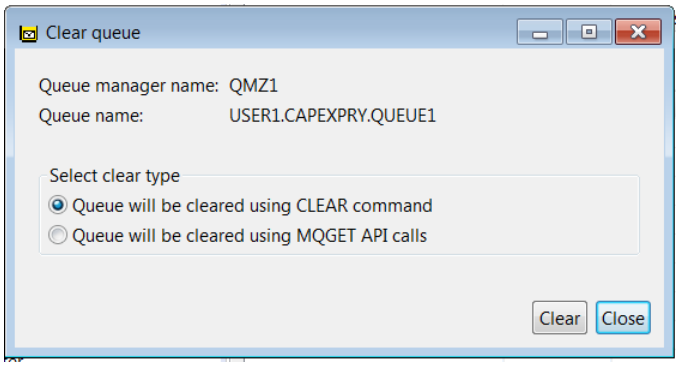
QMZ1 STOP QMGR

____2. Once *QMZ1MSTR* address space has stopped, restart the queue manager by entering that start command.

QMZ1 START QMGR

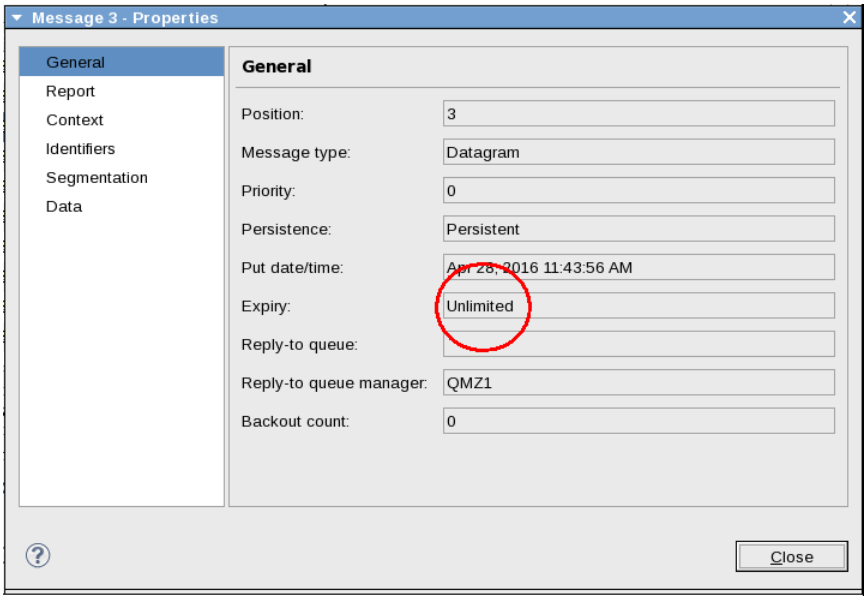
____3. Wait until the two address spaces (*QMZ1MSTR* and *QMZ1CHIN*) are started before continuing.

____4. Any messages on the *USER1.CAPEXPRY.QUEUE1* should be cleared using MQ Explorer.



____5. From the *USER1.CAPEXPRY.JCL* data set, submit the *PUTMSGS* job again.

____6. Browse the new messages and check the properties of one of the messages.



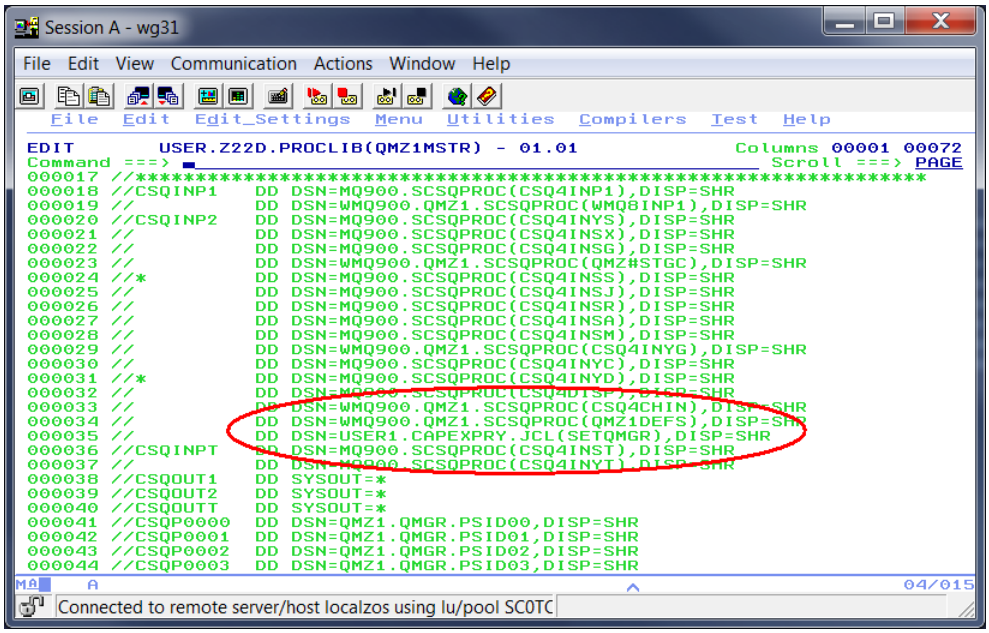
____7. The reason the expiry is back to *Unlimited* is that the command used to enable the option on the queue manager is temporary. It is a 'tuning' option that is only valid for the duration until the queue manager is stopped. In the next step, the queue manager startup will be modified to make this permanent.

Step 6 – Set up the queue manager for CAPEXPRY all the time

____1. From the *Dataset List Utility* panel ***(=3.4***), enter ***USER.PROCLIB*** in the *Dsname Level* field and press **ENTER**.

____2. Edit member *QMZ1MSTR* in data set *USER.PROCLIB.*

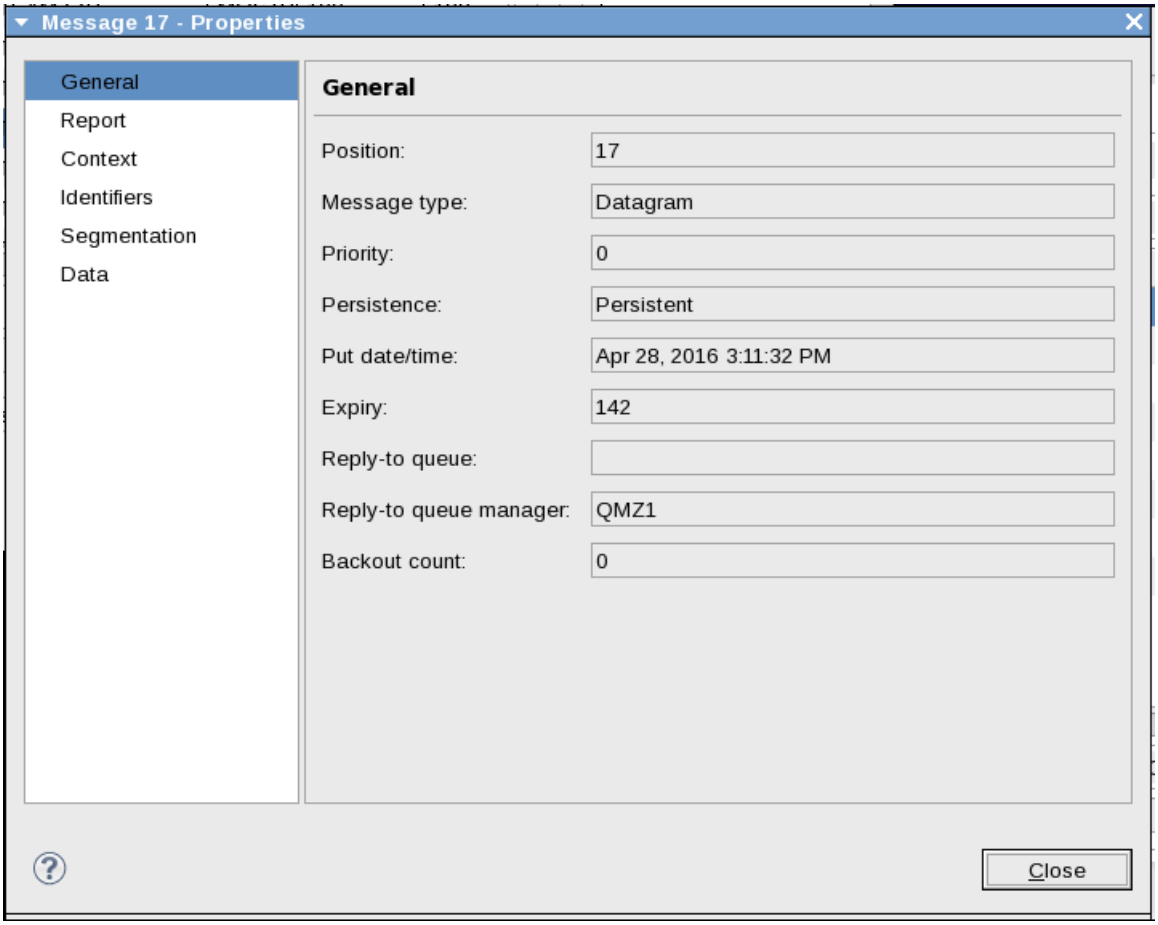
____3. At the end of the *CSQINP2* concatenation, insert a line to include the command to turn on CAPEXPRY for the queue manager from the *USER1.CAPEXPRY.JCL* library as shown. Save the *QMZ1MSTR* member with these changes.



____4. Stop and restart the queue manager as described in the previous step.

____5. Using the *PUTMSGS* job in the *USER1.CAPEXPRY.JCL* PDS, put 10 more messages on the queue.

____6. Using the MQ Explorer, browse the queue – note that you shall probably have to reconnect as the queue manager has been cycled since your last connection. Pick one of the recent messages, and the expiration should be a value other than unlimited.



____7. Congratulations! You have completed the implementation and testing of CAPEXPRY!