Comparing buffers above and below the bar

Lab Objectives

This lab has the following objectives:

1) To compare the use and costs of above and below the bar bufferpool allocation
2) To familiarize administrators with the new parameters

General Lab Information and Guidelines

1. Any time the labels TEAM00 or TEAMXX are used, please replace the '00' or 'XX' with your team ID (TEAM01 – TEAM20).
2. These images were captured on queue managers QML1 and QML3, please use the queue manager assigned.
3. There are four queue managers for use in this workshop. Each team is assigned a primary queue manager as follows:

   - QML1 - TEAM01, TEAM05, TEAM09, TEAM13, TEAM17
   - QML2 - TEAM02, TEAM06, TEAM10, TEAM14, TEAM18
   - QML3 - TEAM03, TEAM07, TEAM11, TEAM15, TEAM19
   - QML4 – TEAM04, TEAM08, TEAM12, TEAM16, TEAM20

2. All teams should signon to the MPX1 LPAR.
3. The passwords for this lab will be provided by the workshop leaders.
4. Any difficulty with connectivity should be reported, but please remember that the connections may be slow.
5. To accommodate limited storage, JCL libraries have been set up for each team as shown below. Please note that you should use the JCL library indicated by TEAMXX where the XX is your team ID.

   TEAM01.BPBAR.JCL
   TEAM02.BPBAR.JCL
   TEAM03.BPBAR.JCL
   TEAM04.BPBAR.JCL
   TEAM05.BPBAR.JCL
   TEAM06.BPBAR.JCL
   TEAM07.BPBAR.JCL
   TEAM08.BPBAR.JCL
   TEAM09.BPBAR.JCL
   TEAM10.BPBAR.JCL…

LAB – Part I – Connecting to your queue manager

In this section, you will connect to your primary queue manager from the MQ Explorer. In the examples shown the connection is being made to QML1. Please use the queue manager assigned above.

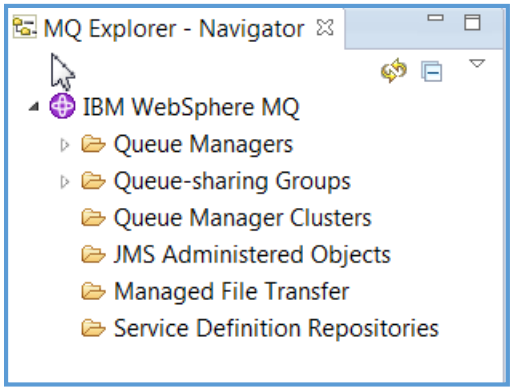
You will notice some differences between the captured images and the directions. They are due to some environmental differences between the VMWare image and running natively. There will also be some notes on features that are not yet working, these items are highlighted as 'TechTips.'

Step 1 – Start the ATS MQ VMware image

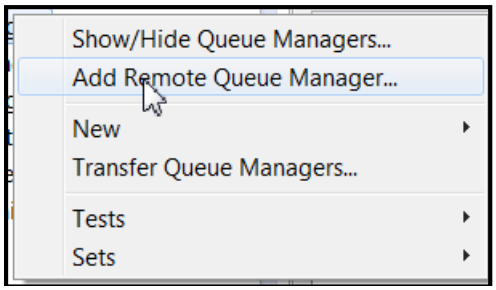
1. If not already started double click on the desktop link to the ATSWMQ VMWare image.
2. The password is that most secure of all passwords, passw0rd.

Step 2 – Defining your primary queue manager to the MQ Explorer

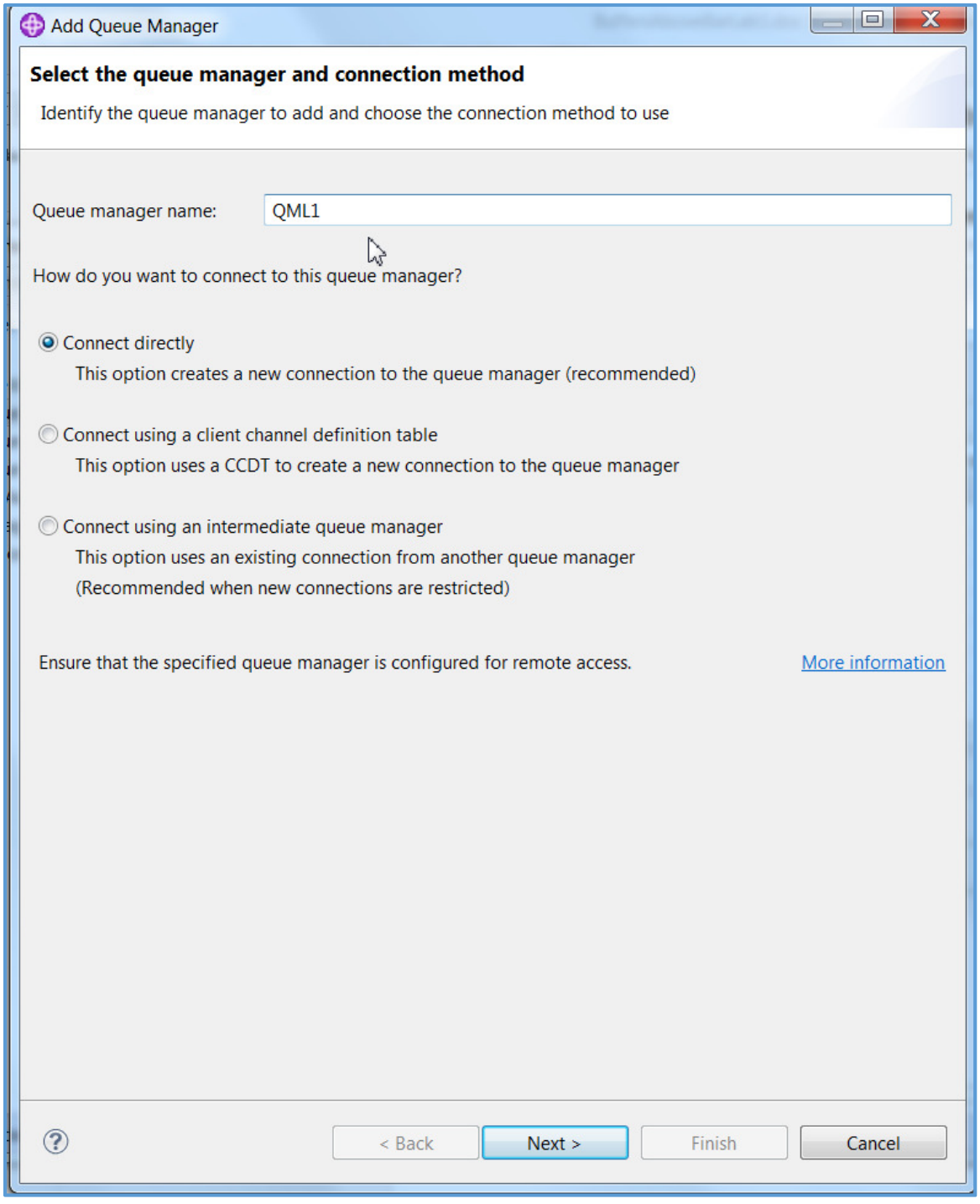
1. Start the MQExplorer . The Navigator pane should look something like this:



2. Right click on the Queue Managers folder and select Add Remote Queue Manager



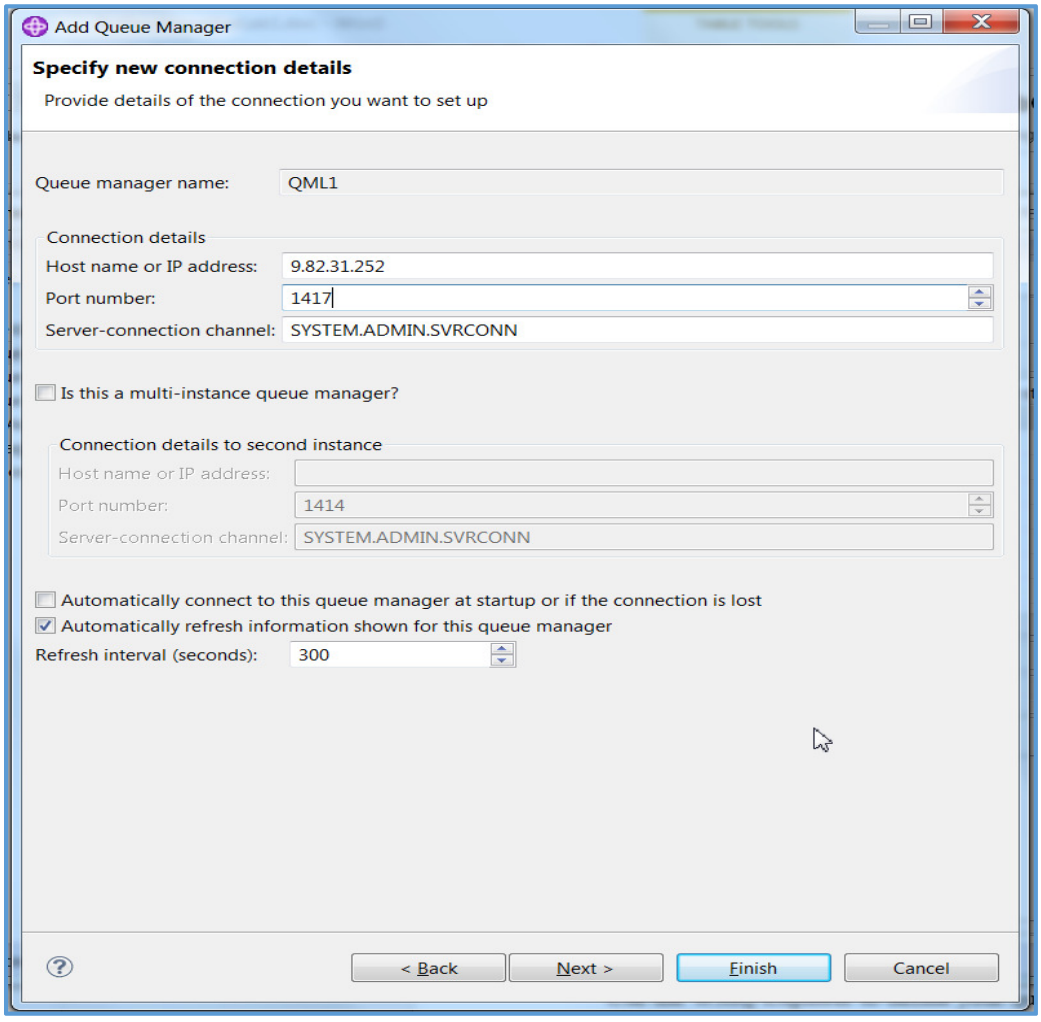
3. On the Add Queue Manager panel, enter your primary queue manager name, make sure that 'Connect directly' is selected, and click on the Next button.



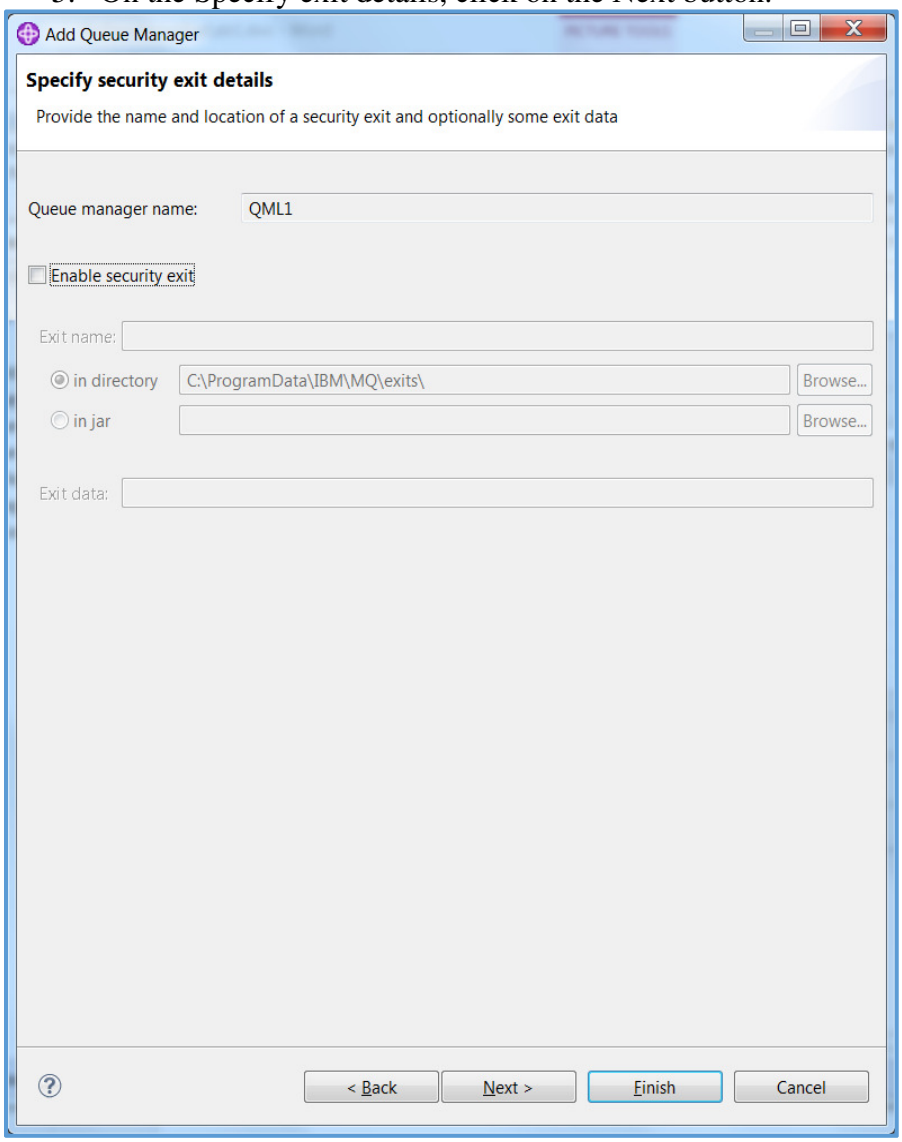
TECHTIP: For the queue manager name and many other fields CASE matters. All the z/OS queue managers use upper case names.

4. In the host name or IP address field and the port number fields enter the values that match your queue manager as shown in this table below. Then click on the 'Next' button.

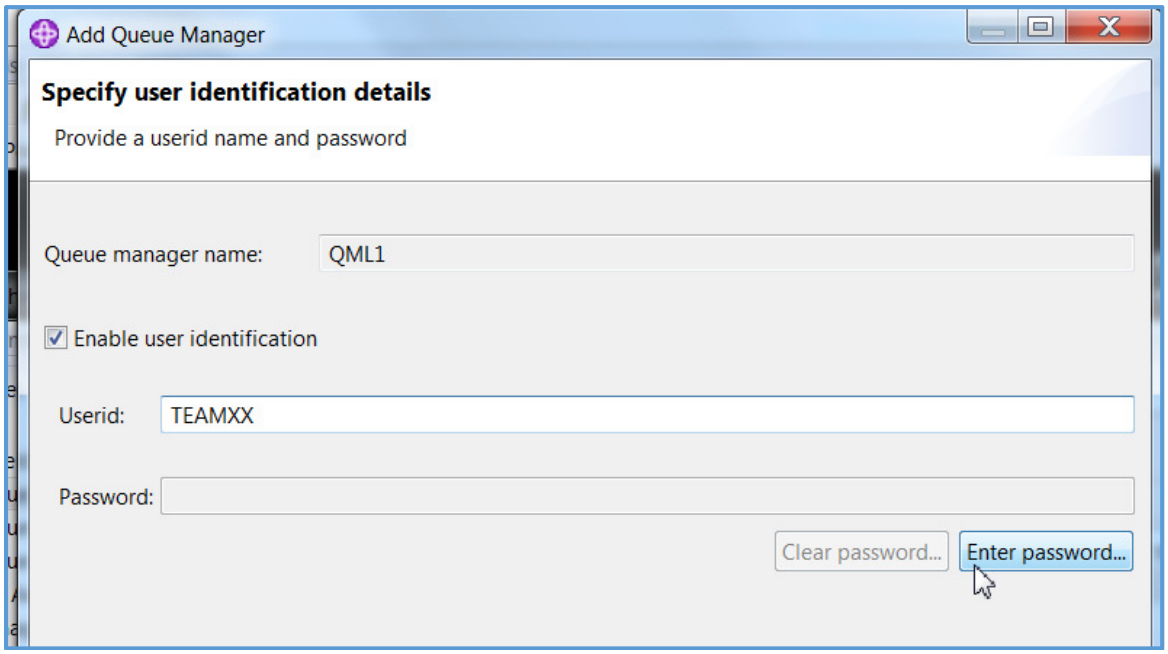
Queue Manager	Host Name	IP address	Port
QML1	mpx1	9.82.31.252	1417
QML2	mpx2	9.82.31.253	1417
QML3	mpx1	9.82.31.252	1418
QML4	mpx2	9.82.31.253	1418



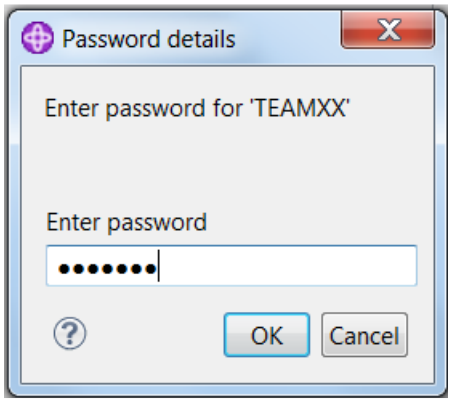
5. On the Specify exit details, click on the Next button.



6. On the ‘Specify user identification details’ panel, enter your team id and click on the ‘Enter password’ button.

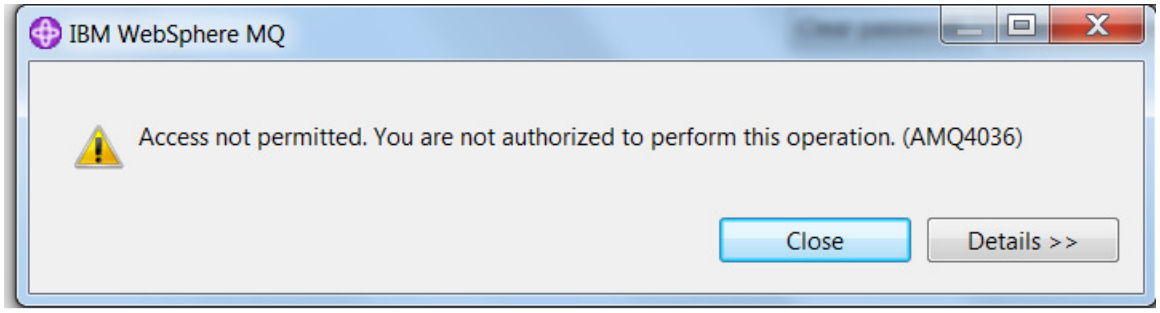


7. On the ‘Enter password’ panel, enter the password ‘b00tcamp’, and click the OK button.

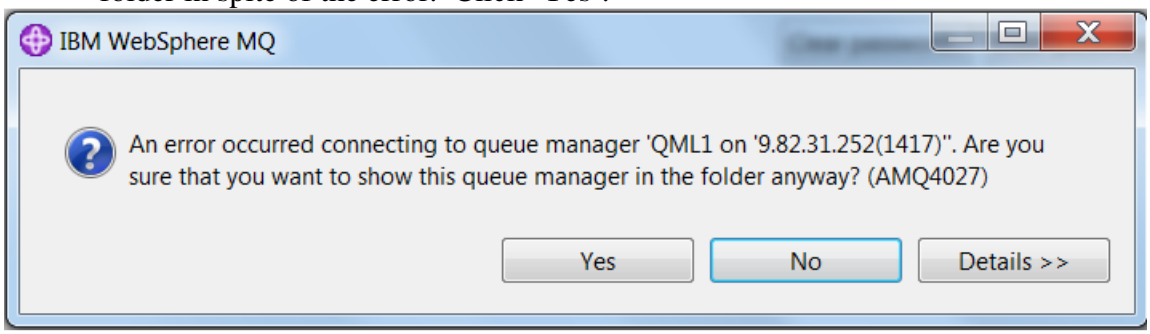


8. Then click the 'Finish' button.

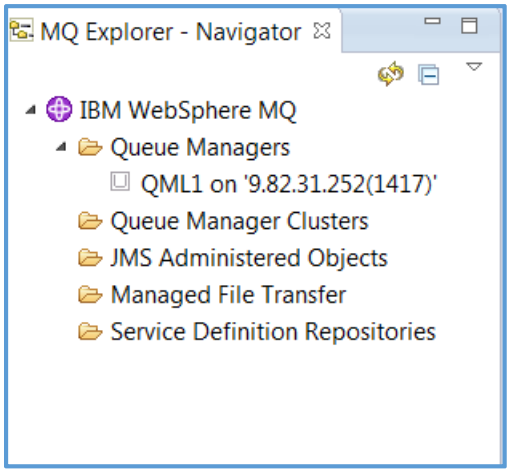
9. You may get a security error, like what is shown here. If you do, click on the Close button.



10. The next message is to verify that you want to add the queue manager to your folder in spite of the error. Click 'Yes'.

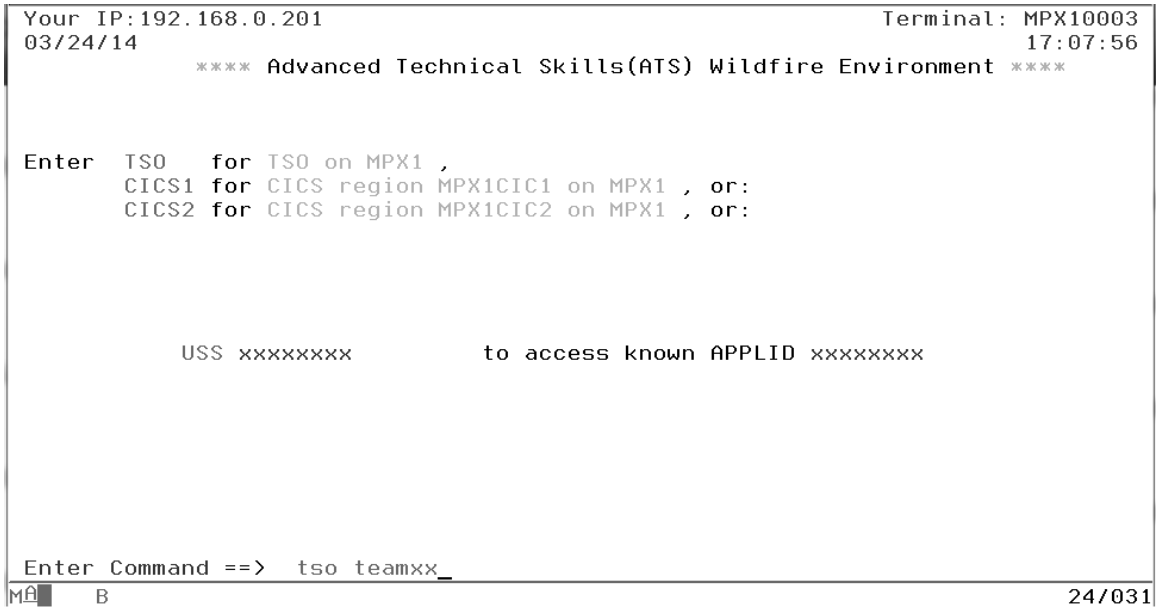


11. The navigator panel should show the queue manager, but in an unconnected state.

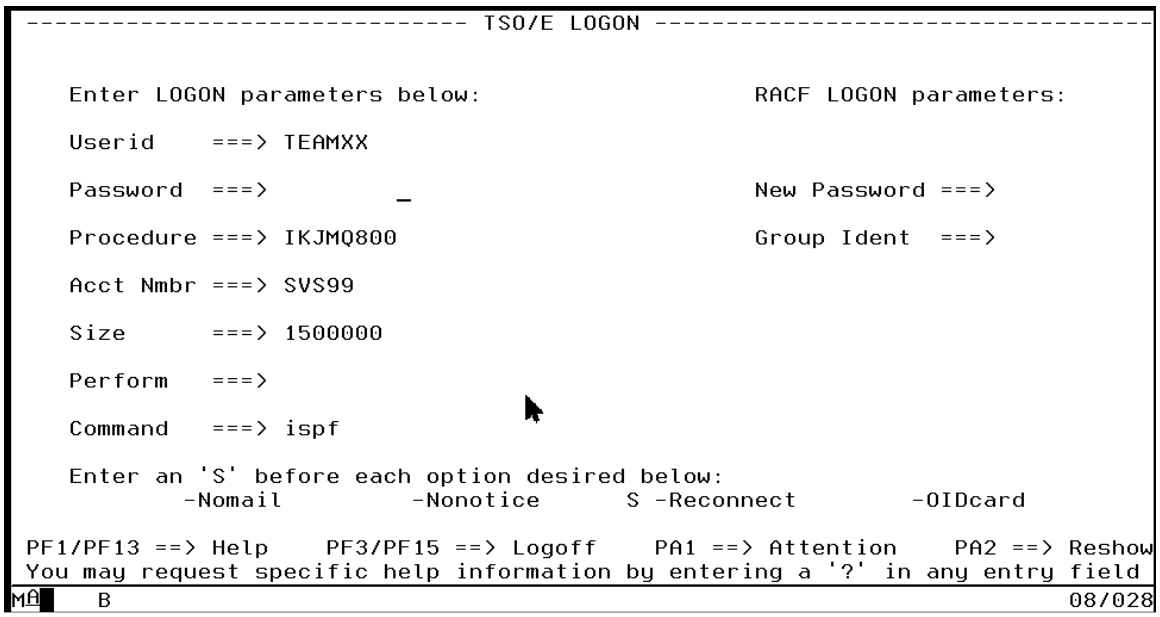


Step 3 – Checking for errors in the queue manager address space

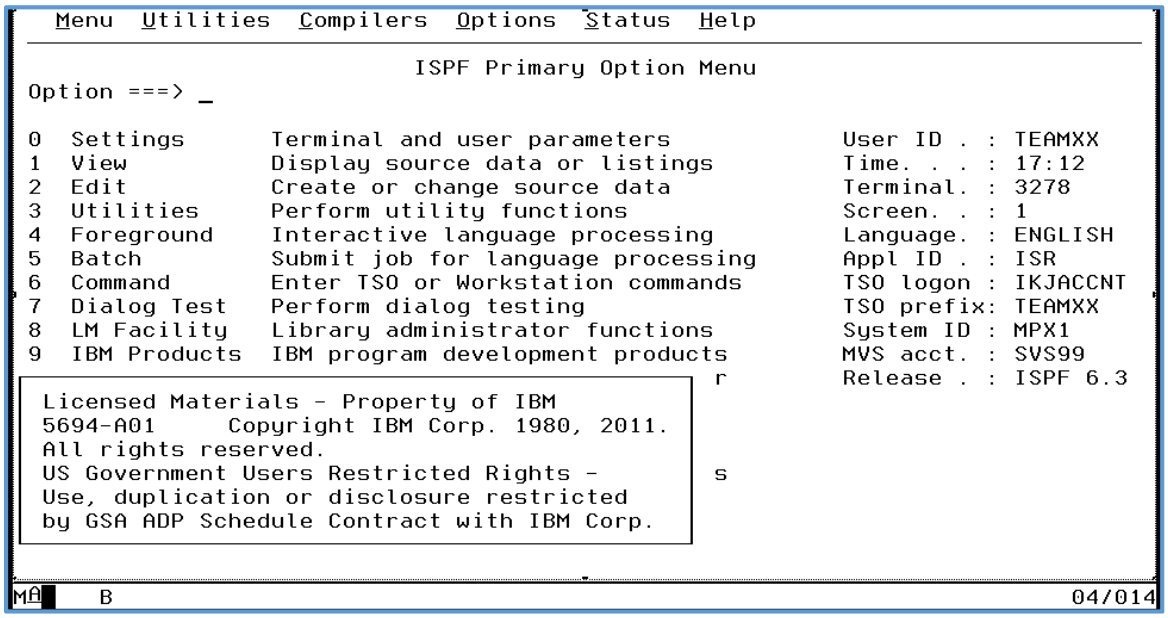
1. To check the error as shown in the queue manager, start the PCOMM session for MPX1 if your primary queue manager is an odd number. If your primary queue manager is an odd number, start the MPX2 session. Even if you do not get the error, please perform these steps to see what is going on in the queue manager. To signon, enter TSO and your team ID as shown (colors are changed to preserve ink):



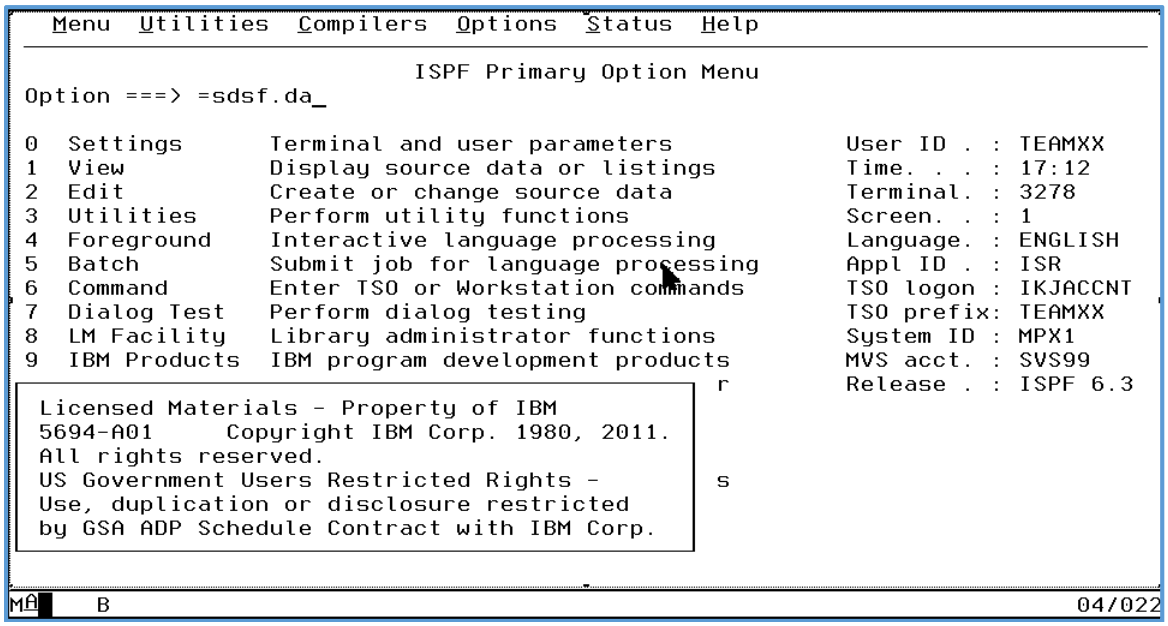
2. Enter your password as shown, and hit the enter key.



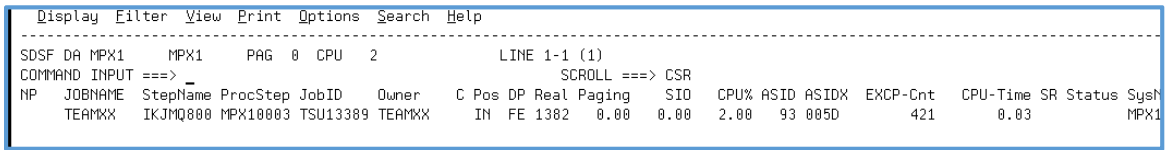
3. If your password is accepted, and it should be, continue hitting the enter key until you see this screen.



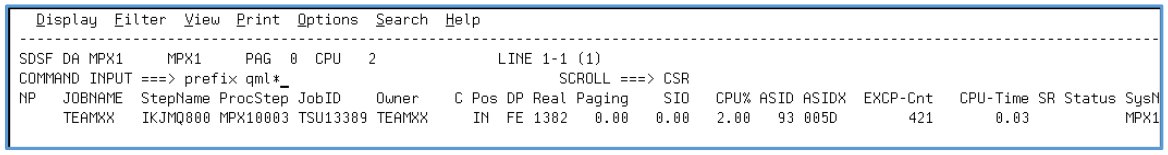
4. On the Option line, type in '=SDSF.DA' without the quotes. This will navigate directly to the active directory of running jobs in the environment.



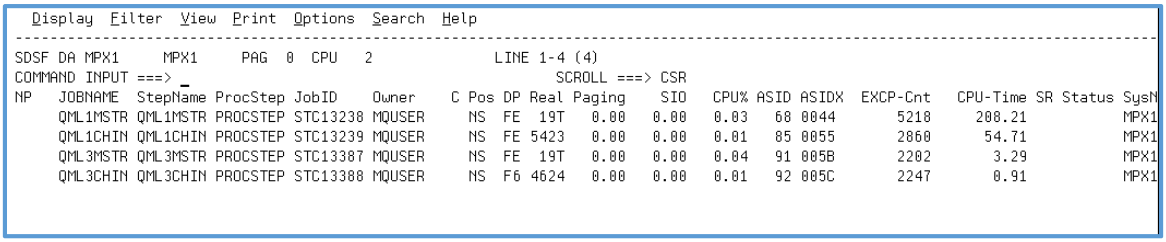
5. If you do not see the queue manager and channel initiator address spaces, as is shown in the example, the prefix must be changed.



6. To change the prefix, enter 'prefix qml*' without the quote on the command input line and hit the enter key.



7. The queue managers and channel initiator address spaces should be shown.



8. Put a question mark beside the queue manager (QML#MSTR, where the # is your primary queue manager number) and press enter to show the different outputs from the job.

```
 Display  Filter  View  Print  Options  Search  Help
------------------------------------------------------------------------------
SDSF DA MPX1     MPX1     PAG  0  CPU   2             LINE 1-4 (4)
COMMAND INPUT ===>                                          SCROLL ===
NP   JOBNAME  StepName ProcStep JobID    Owner    C Pos DP Real Paging
 ?_  QML1MSTR QML1MSTR PROCSTEP STC13238 MQUSER     NS  FE  19T   0.00
     QML1CHIN QML1CHIN PROCSTEP STC13239 MQUSER     NS  FE 5423   0.00
```

9. Select the JESMSGLG as shown below, and once again press enter.

```
 Display  Filter  View  Print  Options  Search  Help
-------------------------------------------------------
SDSF JOB DATA SET DISPLAY - JOB QML1MSTR (STC13238)
COMMAND INPUT ===>
NP   DDNAME   StepName ProcStep DSID Owner    C Dest
 s_  JESMSGLG JES2                2 MQUSER   S
     JESJCL   JES2                3 MQUSER   S
     JESYSMSG JES2                4 MQUSER   S
     CSQOUT1  QML1MSTR          101 MQUSER   S
     CSQOUT2  QML1MSTR          102 MQUSER   S
     CSQOUTT  QML1MSTR          103 MQUSER   S
```

10. The job log should be displayed, as shown.

```
  Display  Filter  View  Print  Options  Search  Help
-------------------------------------------------------------------------------
SDSF OUTPUT DISPLAY QML1MSTR STC13238  DSID     2 LINE 0       COLUMNS 02- 133
COMMAND INPUT ===> _                             SCROLL ===> CSR
******************************** TOP OF DATA ********************************
                  J E S 2  J O B  L O G  --  S Y S T E M  M P X 1  --  N O D E  W S C 3 0 0

10.15.04 STC13238 ---- TUESDAY,   18 MAR 2014 ----
10.15.04 STC13238  IEF695I START QML1MSTR WITH JOBNAME QML1MSTR IS ASSIGNED TO USER MQUSER  , GROUP SYS1
10.15.04 STC13238  $HASP373 QML1MSTR STARTED
10.15.04 STC13238  IEF403I QML1MSTR - STARTED - TIME=10.15.04
10.15.04 STC13238  CSQY000I QML1 IBM WebSphere MQ for z/OS VD04MAR14
10.15.04 STC13238  CSQY001I QML1 QUEUE MANAGER STARTING, USING PARAMETER MODULE CSQZPARM
10.15.04 STC13238  CSQ3111I QML1 CSQYSCMD - EARLY PROCESSING PROGRAM IS VD04MAR14 LEVEL  424
   424             007-000
10.15.04 STC13238  CSQY100I QML1 SYSTEM parameters ...
10.15.04 STC13238  CSQY101I QML1 LOGLOAD=500000, OPMODE=(NEWFUNC ,800)
10.15.04 STC13238  CSQY102I QML1 CMDUSER=CSQOPR, QMCCSID=0, ROUTCDE=( 1)
10.15.04 STC13238  CSQY103I QML1 SMFACCT=NO (00000000), SMFSTAT=NO (00000000), STATIME=30
10.15.04 STC13238  CSQY104I QML1 OTMACON=  429
   429             (,,DFSYDRU0,2147483647,CSQ)
10.15.04 STC13238  CSQY105I QML1 TRACSTR=( 1), TRACTBL=99, CONNSWAP=YES
10.15.04 STC13238  CSQY106I QML1 EXITTCB=8, EXITLIM=30, WLMTIME=30, WLMTIMU=MINS
10.15.04 STC13238  CSQY107I QML1 QSGDATA=(QSGM,DSNXPLEX,DSNX,4,4)
10.15.04 STC13238  CSQY108I QML1 RESAUDIT=YES, QINDXBLD=WAIT, CLCACHE=STATIC
```

11. Navigate to the bottom of the log to see the most recent messages my putting an ‘m’ (without the quotes) in the command input line and pressing the F8 key.

The error message associated with your attempt should be at or close to the bottom of the log.

```
 SDSF OUTPUT DISPLAY QML1MSTR STC13238  DSID     2 LINE 2,248   COLUMNS 02- 133
 COMMAND INPUT ===> _                                          SCROLL ===> CSR
16.25.05 STC13238  CSQY220I QML1 CSQSCTL Queue manager storage usage:  765
   765             local storage: used 415MB, free 1323MB: above bar: used 690MB, free
   765             1GB
16.44.29 STC13238  ICH408I USER(TEAMXX  ) GROUP(SYS1    ) NAME(GENERAL             )  771
   771               LOGON/JOB INITIATION - INVALID PASSWORD
16.44.29 STC13238  IRR013I  VERIFICATION FAILED. INVALID PASSWORD GIVEN.
17.25.05 STC13238  CSQY220I QML1 CSQSCTL Queue manager storage usage:  779
   779             local storage: used 415MB, free 1323MB: above bar: used 690MB, free
   779             1GB
****************************** BOTTOM OF DATA ********************************************************
```

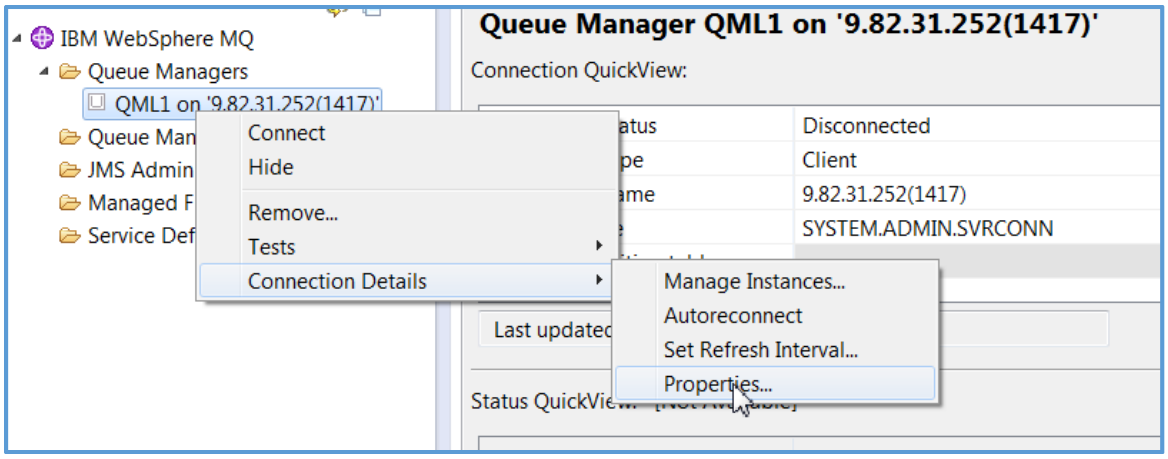
12. Why do you think the password was invalid? Hint: remember the TechTip? This environment does not do uppercase translation, as some might. The next steps will correct the connection information.

TECHTIP 2: User ID and password verification against RACF, or another ESM, is now done automatically when sent to the queue manager. This behavior may be different from what customers have seen in the past.

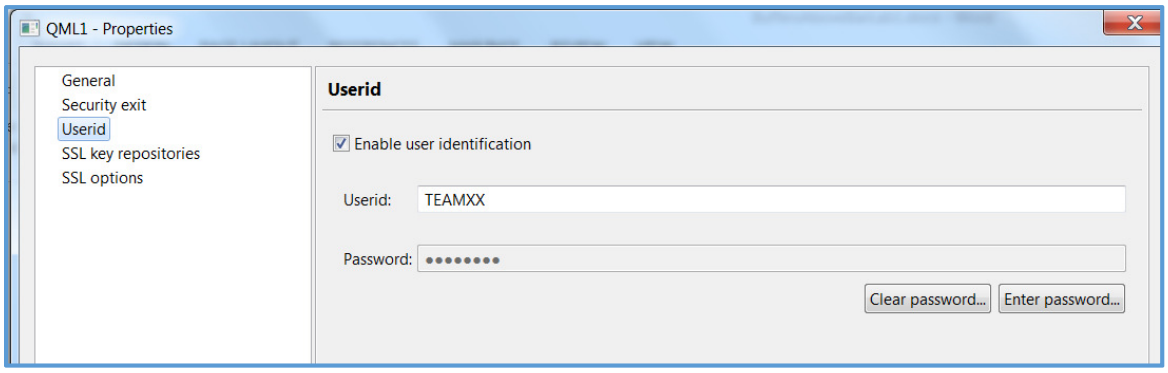
TECHTIP 3: At the time this lab was created, the delivered sample security exit, CSQ4BCX3, causes an abend when used. Specifically, the following is seen in the JES jog:
10.13.55 STC13337 +CSQX111E QML3 CSQXDISP User channel exit error, TCB=008CC090 058
058 reason=5C6000-00E70FFF
10.13.55 STC13337 +CSQX599E QML3 CSQXRESP Channel SYSTEM.ADMIN.SVRCONN ended abnormally
This exit was used in V7.1 to do the user ID and password checking. This has been reported to the development lab.

Step 4 – Repairing the connection information – may not be required

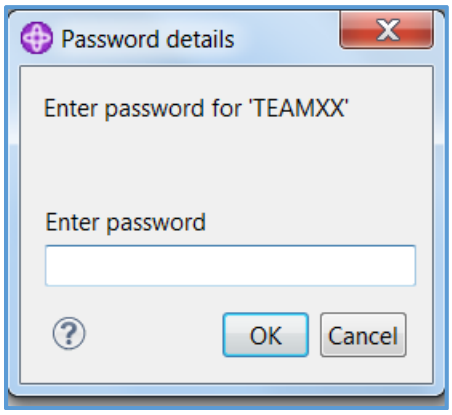
1. Return to the MQ explorer and right click on the queue manager name, select Connection Details, then Properties.



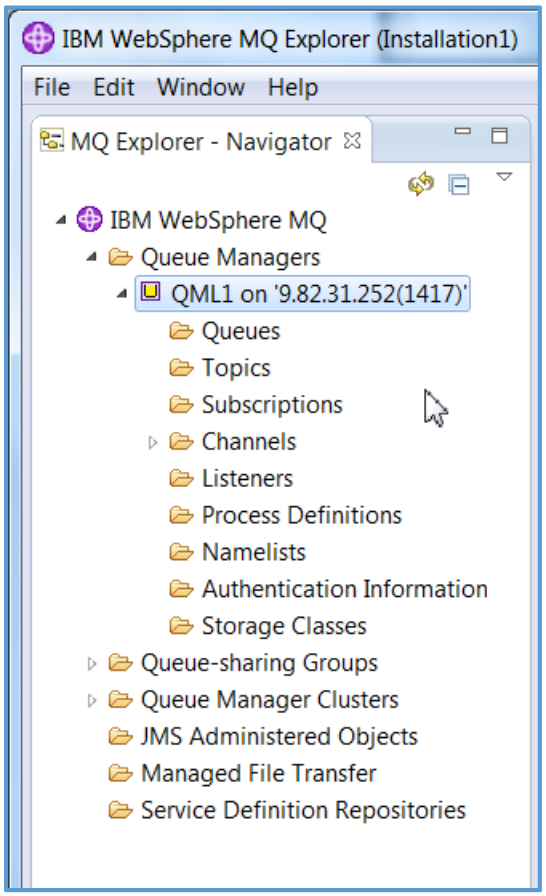
2. On the Queue manager properties display, select the Userid tab, then click on the Enter password button.



3. Enter the password on the Password details pane in UPPER CASE and click the OK button.



4. Click the OK button when the Properties panel has returned.
5. Right click on the queue manager name, and select 'Connect'
6. The connection should work properly at this point. The panel should show the connected queue manager, and you should then be able to expand that to show the resource definitions.



LAB – Part II - Defining your queues

Use the WMQ Explorer to define your queues for these exercises. If your explorer session has been shut down, please restart it and connect to your primary queue manager.

For this test, two queues need to be defined, one on each predefined storage class. These are based on your team ID as shown:

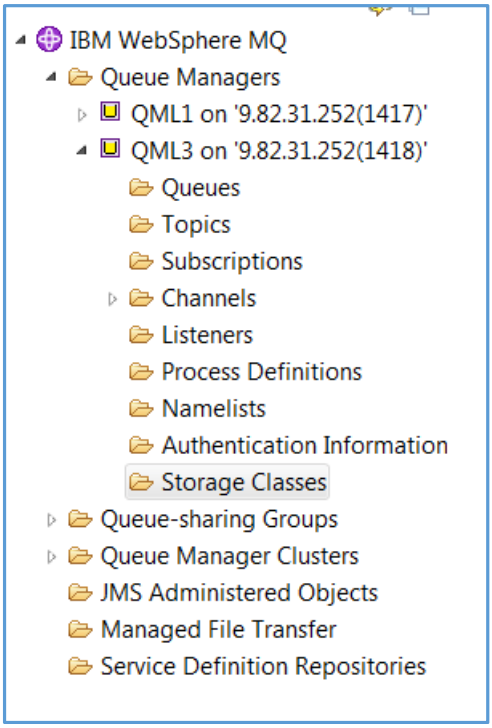
Team	Queue Manager	Below the Bar	Above the Bar
TEAM01	QML1	STGCLS10	STGCLS11
TEAM02	QML2	STGCLS10	STGCLS11
TEAM03	QML3	STGCLS10	STGCLS11
TEAM04	QML4	STGCLS10	STGCLS11
TEAM05	QML1	STGCLS12	STGCLS13
TEAM06	QML2	STGCLS12	STGCLS13
TEAM07	QML3	STGCLS12	STGCLS13
TEAM08	QML4	STGCLS12	STGCLS13
TEAM09	QML1	STGCLS14	STGCLS15
TEAM10	QML2	STGCLS14	STGCLS15
TEAM11	QML3	STGCLS14	STGCLS15
TEAM12	QML4	STGCLS14	STGCLS15
TEAM13	QML1	STGCLS16	STGCLS17
TEAM14	QML2	STGCLS16	STGCLS17
TEAM15	QML3	STGCLS16	STGCLS17
TEAM16	QML4	STGCLS16	STGCLS17
TEAM17	QML1	STGCLS18	STGCLS19
TEAM18	QML2	STGCLS18	STGCLS19
TEAM19	QML3	STGCLS18	STGCLS19
TEAM20	QML4	STGCLS18	STGCLS19

Step 1 – Verify the storage classes

1) The storage classes can be displayed from the MQ Explorer, but that does not give all the critical bit of information about the location of the bufferpool the storage class uses.

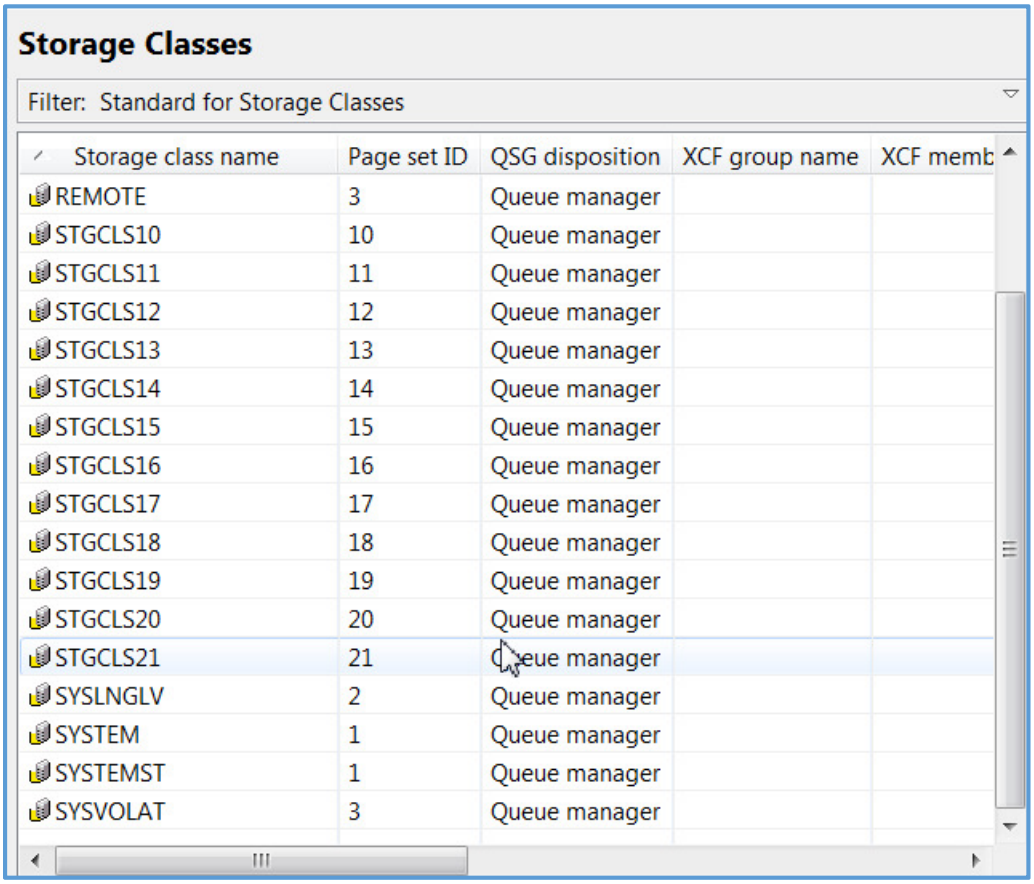
TECHTIP 4: The storage class display has never shown the association with a bufferpool. There has also never been a Bufferpool display from the MQ Explorer, or a 'Display Usage' capability that would show the relationship between the STGCLASS and bufferpool. To get this information, you have to use the commands in z/OS and review the JES log.

2) From the queue manager resource list in the Navigator pane, click on the Storage Classes folder to display the currently defined classes.



TECHTIP 5: MQ V8 increased the number of available bufferpools to 99, the same number of pagesets available. For customers concerned about performance and isolation of resource use, in particular those customers using QREP, defining a one-to-one relationship between the bufferpool and pageset is recommended for application queues. That relationship helps identify problem area more quickly, and in some cases, move resources around to alleviate temporary performance and capacity issues more easily.

3) The storage class display from the V8 version of the MQ Explorer looks as shown:



4) Return to the TSO session, and in the JES message log of the queue manager address space of your primary queue manager, enter the command:

   /cpf display usage

   Where cpf is the command prefix string for the queue manager. In the ATS environment the cpf is the subsystem ID. (QML1, QML2, QML3, or QML4)

```
 SDSF OUTPUT DISPLAY QML3MSTR STC13395  DSID     2 LINE 376     COLUMNS 02- 133
 COMMAND INPUT ===> /qml3 display usage_                      SCROLL ===> CSR
02.45.46 STC13395  CSQY220I QML3 CSQSCTL Queue manager storage usage:  189
   189             local storage: used 419MB, free 1319MB: above bar: used 701MB, free
   189             1GB
```

5) The display will look something like this:

```
08.30.28 STC13395  CSQI010I QML3 Page set usage ...  218
   218               Page Buffer     Total    Unused  Persistent  NonPersist Expansion
   218                set   pool     pages     pages  data pages  data pages        count
   218             _    0      0      1078      1040          38           0 USER       0
   218             _    1      0      1078      1046          32           0 USER       0
   218             _    2      1      1078      1076           2           0 USER       0
   218             _    3      2      1078      1078           0           0 USER       0
   218             _    4      3      1078      1069           9           0 USER       0
   218             _   10     10      2698      2698           0           0 USER       0
   218             _   11     11      2698      2698           0           0 USER       0
   218              End of page set report
08.30.28 STC13395  CSQI065I QML3 Buffer pool attributes ...  219
   219               Buffer  Available  Stealable   Stealable  Page      Location
   219                 pool    buffers    buffers  percentage  class
   219             _      0      50000      49949          99  4KB       BELOW
   219             _      1      20000      19999          99  4KB       BELOW
   219             _      2      50000      49994          99  4KB       ABOVE
   219             _      3      20000      19975          99  4KB       BELOW
   219             _     10       1000        999          99  4KB       BELOW
   219             _     11       1000        999          99  4KB       ABOVE
   219             _     12       1000          0           0  4KB       BELOW
   219             _     13       1000          0           0  4KB       ABOVE
```

What may be different on the display is the number of stealable buffers and percentage. If no queues have been defined and used yet in the bufferpool these values will be different. Please also note the Page class and location values, these are new for V8.

TECHTIP 6: From the MQ V8 InfoCenter on the Location value:

LOCATION(LOC)(BELOW or ABOVE)
The LOCATION or LOC parameter specify where the memory used by the specified buffer pool is located. LOCATION and LOC are synonyms and either, but not both, can be used. This memory location can be either ABOVE (64 bit) or BELOW (31 bit) the bar. Valid values for this parameter are BELOW or ABOVE, with BELOW being the default. ABOVE can only be specified if OPMODE(NEWFUNC, 800) is in effect. BELOW can be specified regardless of OPMODE(NEWFUNC, 800) being used and has the same effect as not specifying the LOCATION parameter.

When altering a buffer pool care should be taken to make sure there is sufficient storage available if increasing the number of buffers or changing the LOCATION value.

TECHTIP 7: From the MQ V8 InfoCenter on the Page class value:

PAGECLAS(4KB or FIXED4KB)
Optional parameter that describes the type of virtual storage pages used for backing the buffers in the buffer pool.

This attribute applies to all buffers in the buffer pool, including any that are added later as a result of using the ALTER BUFFPOOL command. The default value is 4 KB, which means that pageable 4 KB pages are used to back the buffers in the pool.

4 KB is the only valid value if the buffer pool has its location attribute set to BELOW. If the buffer pool has its LOCATION attribute set to ABOVE, it is also possible to specify FIXED4KB. This means that fixed 4 KB pages, which are permanently in real storage and will never be paged out to auxiliary storage, are used to back the buffers in the buffer pool.

FIXED4KB can only be specified if OPMODE(NEWFUNC, 800) is in effect whereas 4 KB can be specified regardless of the setting of OPMODE(NEWFUNC, 800).

The PAGECLAS attribute of a buffer pool can be altered at any time. However, the alteration only takes place when the buffer pool switches location from above the bar, to below the bar, or the other way round. Otherwise, the value is stored in the log of the queue manager and is applied when the queue manager next restarts.

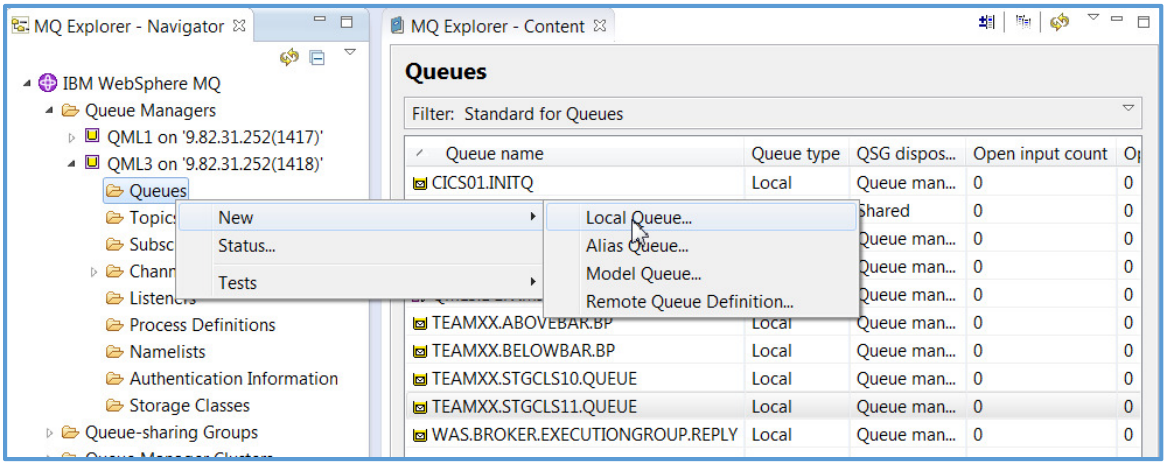
When you specify PAGECLAS(FIXED4KB) the whole buffer pool is backed by page-fixed 4 KB pages, so ensure that there is sufficient real storage available on the LPAR. Otherwise, the queue manager might not start up, or other address spaces might be impacted; for more information, see Address space storage.

See WebSphere MQ Support Pac MP16: WebSphere MQ for z/OS - Capacity planning & tuning for advice on when to use the FIXED4KB value of the PAGECLAS attribute.

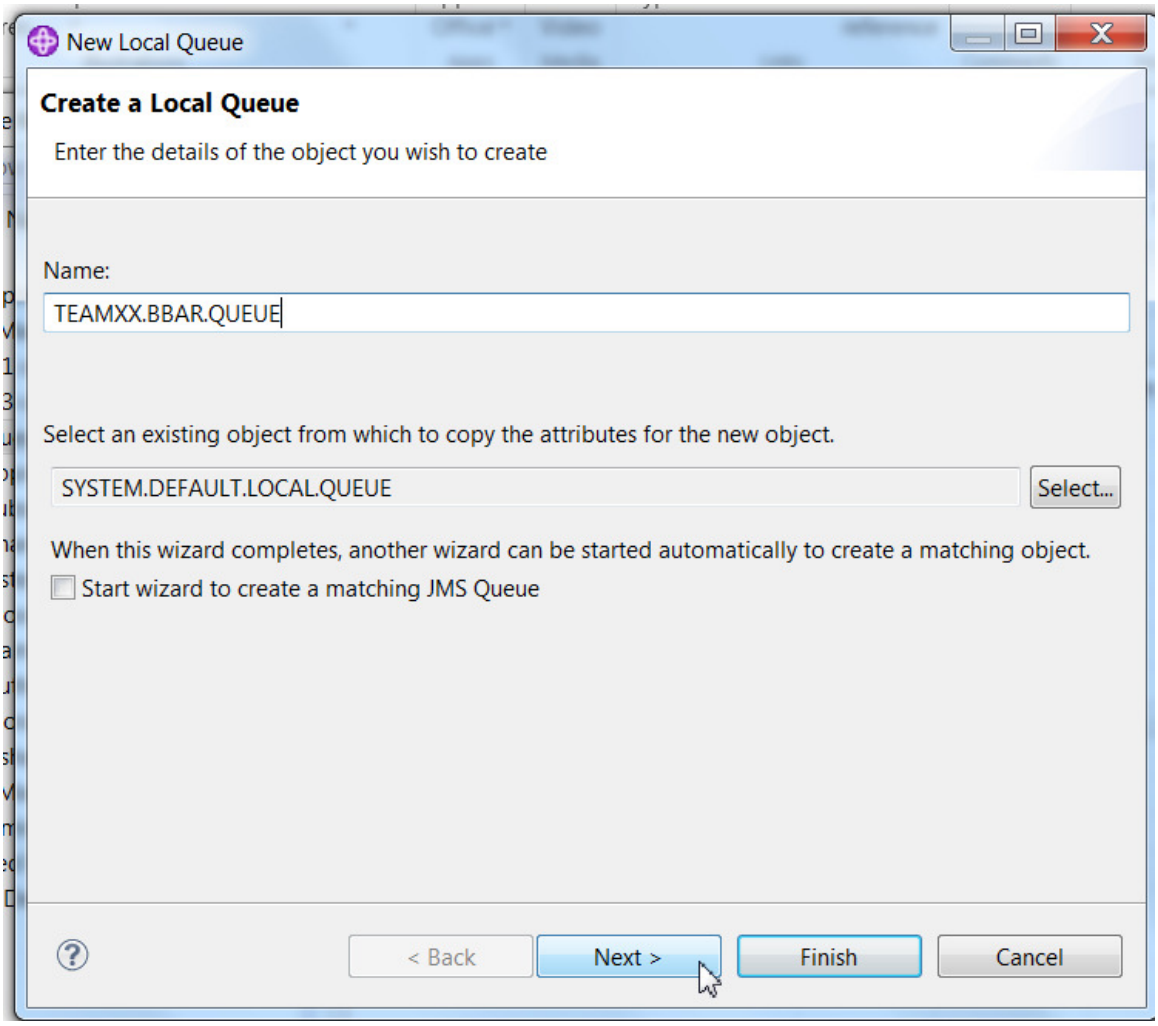
Please note that the bufferpools allocated for the lab are not fixed, as we do not want to have real storage issues.

6) Verify from the display that the even numbered bufferpool defined for your test is below the bar, and the odd numbered bufferpool is above. Please see the table on following the heading LAB – Part II - Defining your queues on page 18.

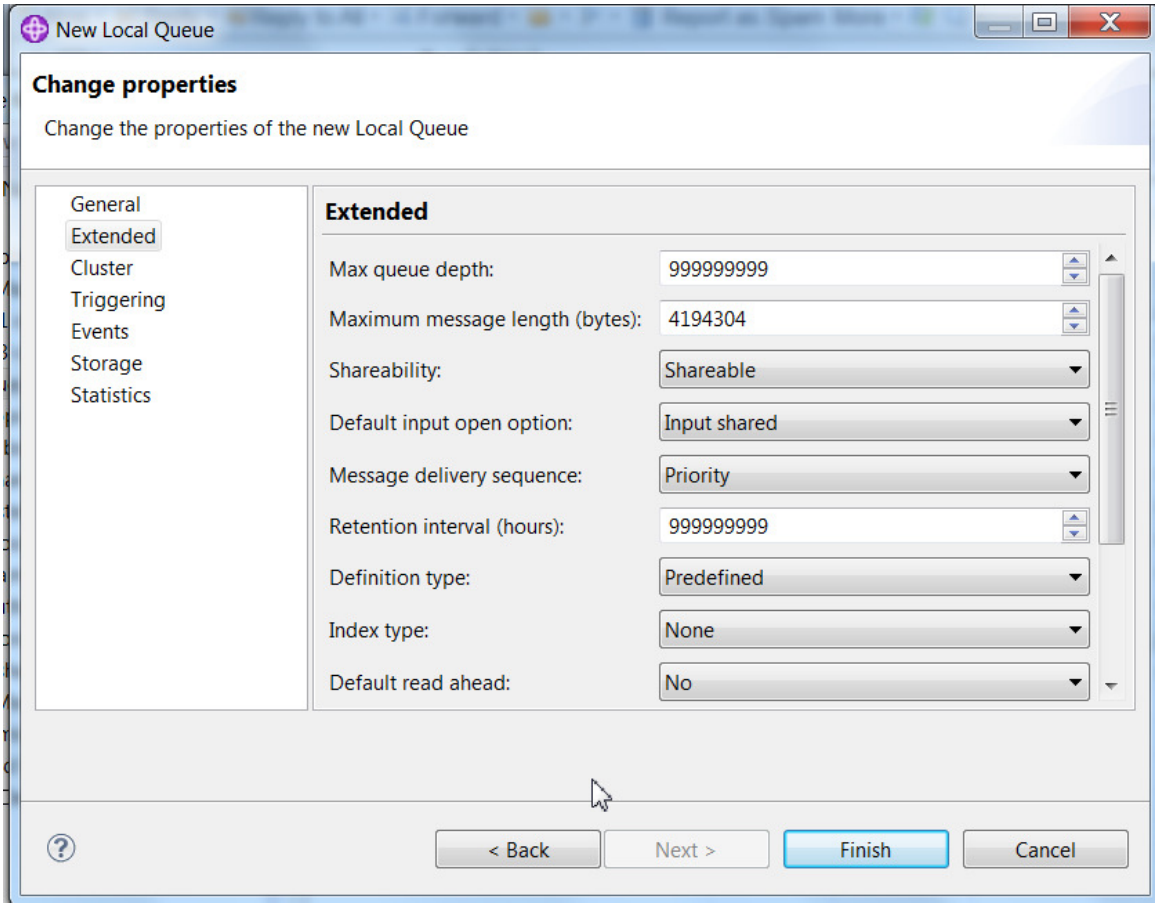
7) To define the below the bar queue, right click on the ‘Queues’ folder on the MQ explorer navigator pane for your queue manager and select ‘New’ the ‘Local queue’ as shown.



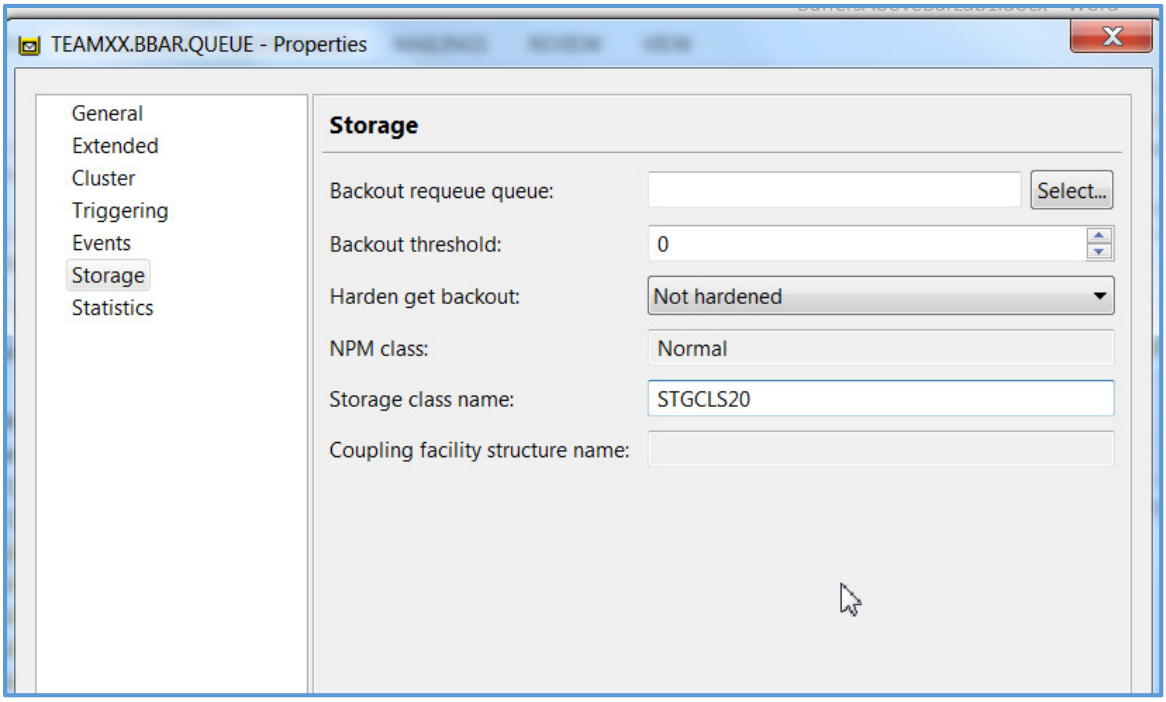
8) Enter the queue name, please use all caps, as TEAMXX.BBAR.QUEUE, where the 'XX' is replaced with your team number. Then click on 'Next.



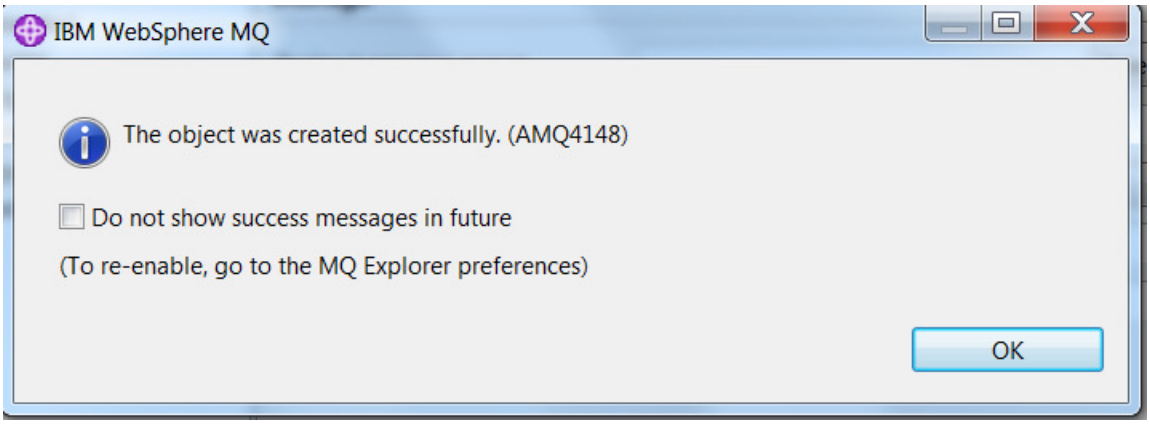
9) Select the ‘Extended’ tab and change the default ‘Shareability’ and ‘Default input open option’ values to ‘Shareable’ and ‘Input shared’ respectively. Then select the ‘Storage’ tab.



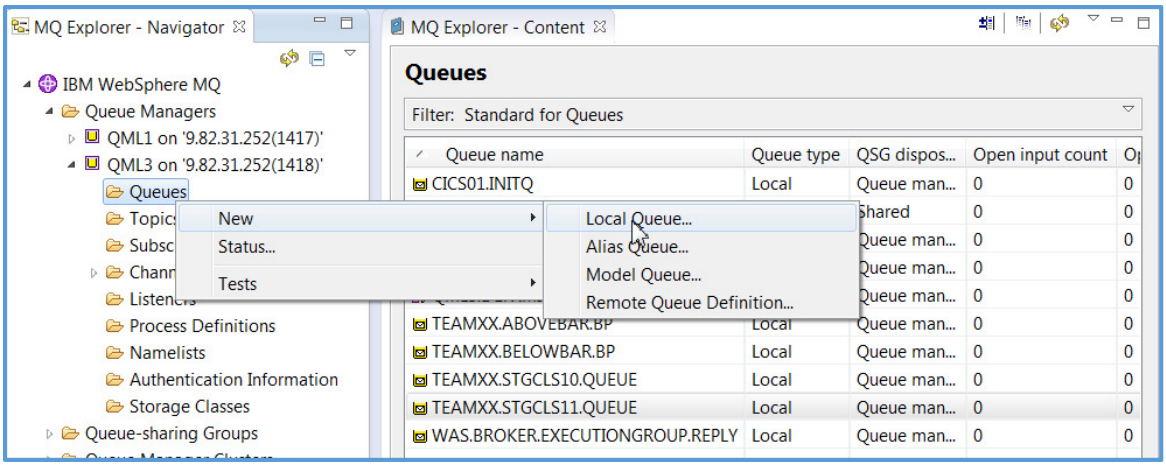
10) Replace the Storage class name ‘DEFAULT’ with the storage class name for the below the bar class defined for your team ID. The example shows STGCLS20, yours will be different. Then click on the ‘Finish’ button.



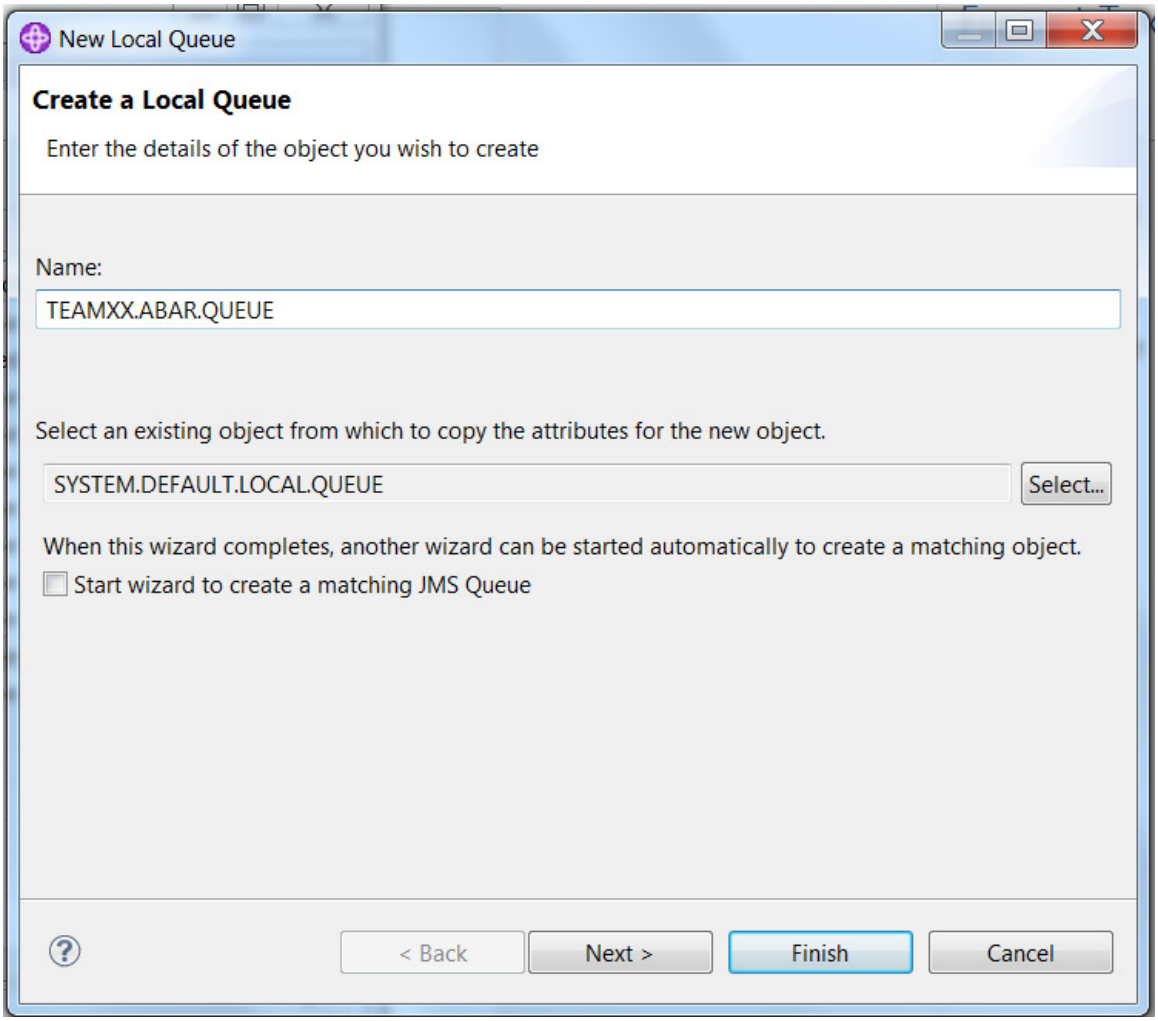
11) The object defined successfully message should be displayed. You can turn it off so it is not displayed again, or leave it on as you prefer.



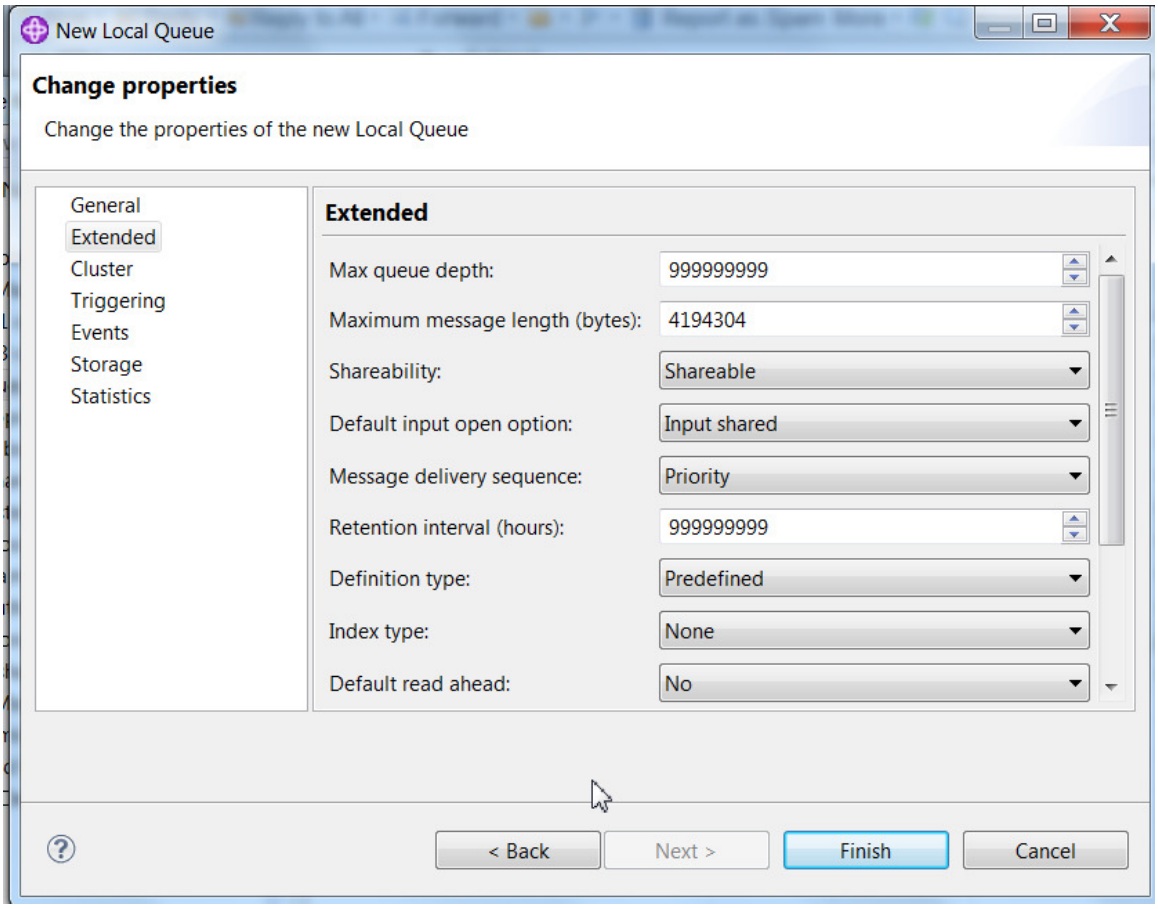
12) To define the above the bar queue, right click on the ‘Queues’ folder on the MQ explorer navigator pane for your queue manager and select ‘New’ the ‘Local queue’ as shown.



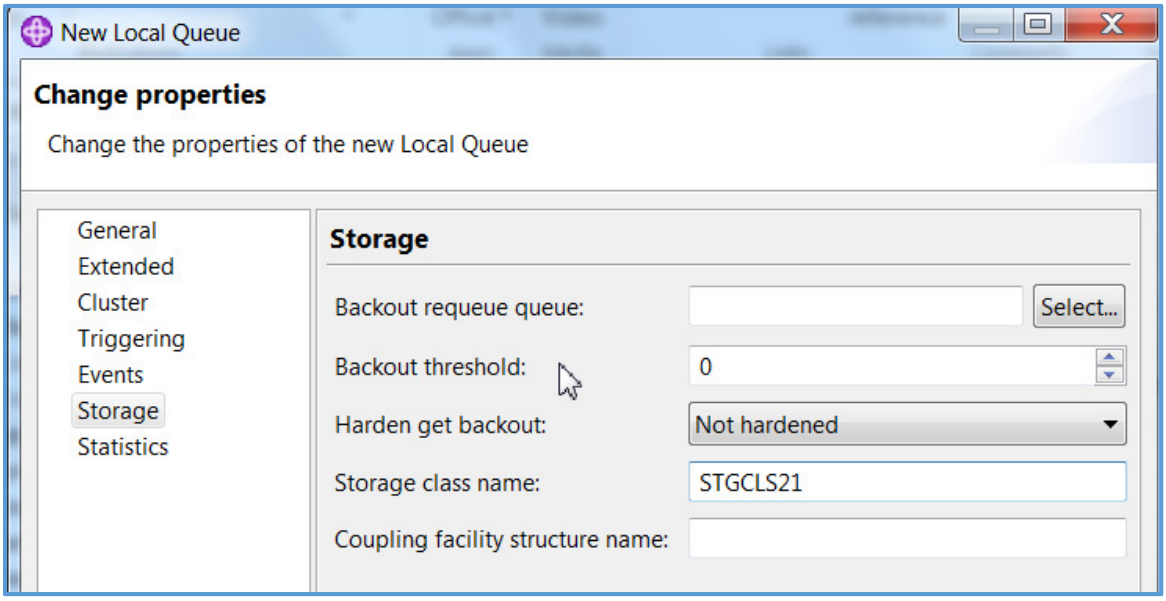
13) Use the queue name TEAMXX.ABAR.QUEUE' replacing the XX with the team number assigned, and click the 'Next' button.



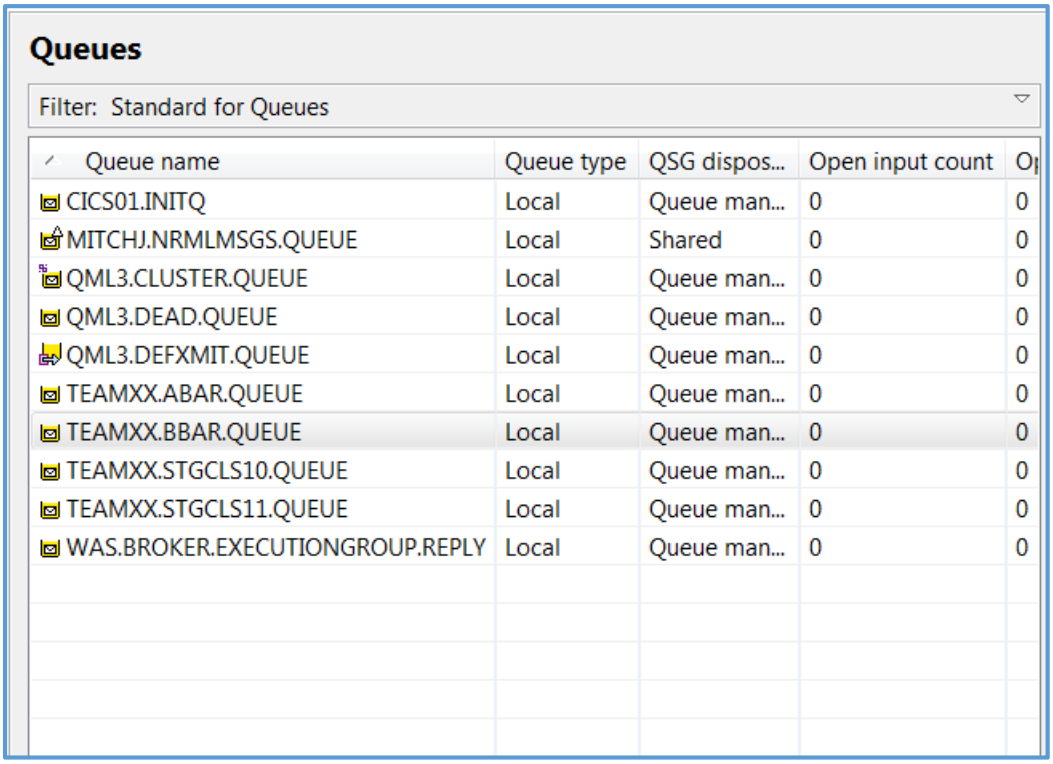
14) On the ‘Extended’ tab, alter the default ‘Shareability’ and ‘Default input open option’ values to ‘Shareable’ and ‘Input shared’ respectively. Then select the ‘Storage’ tab.



15) Replace the Storage class name 'DEFAULT' with the storage class name for the above the bar class defined for your team ID. The example shows STGCLS21, yours will be different. Then click on the 'Finish' button.



16) The queue list should now include the two newly defined queues.



LAB – Part III – Testing and comparing the bufferpools

The key advantage of above the bar bufferpools is the ability to hold many more messages in memory, avoiding I/O to the pagesets. In this exercise we are not able to demonstrate that, but we are focusing on the comparison of runtime costs. Above the bar addressing can be slightly more expensive in CPU costs, but far less expensive than I/O!

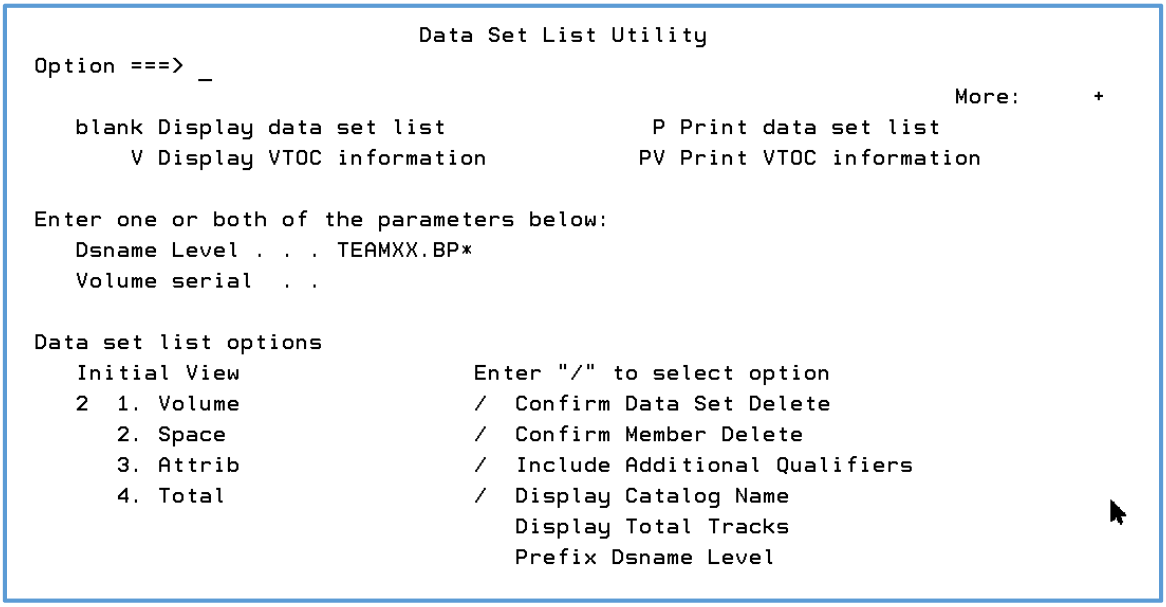
For anyone with experience of DB2 moving buffers above the bar, the initial implementation was reportedly much more expensive than below the bar. MQ for z/OS development has learned from that experience, and has the advantage of newer versions of the operating system.

These tests are designed to compare the costs, and more importantly give some sample tests that customer can reproduce in their environments. It uses the IP13 SupportPac to provide the sample programs, the older version of the MP16 SupportPac to evaluate the MQ SMF data (we will be using canned data, not running the jobs), and standard MQ commands.

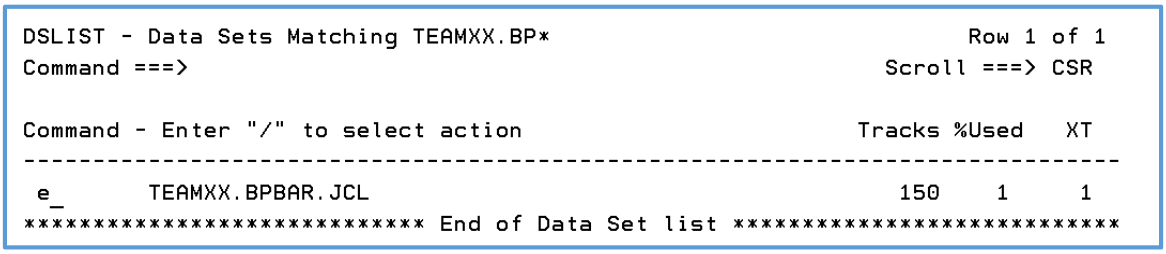
Two sets of tests will be run. The first will compare the run characteristics of below and above bufferpools where no I/O has to be done. The second will compare them when I/O does take place. We would caution everyone running these tests, the numbers presented were gathered when the environment was not being used for any other testing. Your results may (probably will!) vary. A major difference in performance characteristics is when z/OS paging occurs in the environment, a situation that we in ATS have observed from time to time.

Step 1 – Customizing the JCL and Submitting the jobs

1) In the TSO session enter =3.4 in the command line to navigate to the Data Set List Utility panel, and enter TEAMXX.BP*, replacing the XX with your team number in the Dsname level field. Hit the enter key to display the list of data sets.



2) Edit the 'TEAMXX.BPBAR.JCL' data set by entering a 'e' in the command column as shown.



3) Select the V8NOIO10 member and apply the following changes:
   a) Change the 'TEAM??' value to your team number for all occurrences.

```
EDIT       TEAMXX.BPBAR.JCL(V8NOIO10) - 01.07                    Columns 00001 00080
Command ===> c TEAM?? TEAMXX all_                                   Scroll ===> CSR
****** ********************************* Top of Data ***********************************
000001 //TEAM??IO JOB (????,????),'BBAR No IO',NOTIFY=????????
000002 //*
000003 //* THE FIRST STEP IS TO LOAD 300 MESSAGES ON A QUEUE THAT SHOULD NOT GET
000004 //* WRITTEN TO THE PAGESET, IN BUFFERPOOL ONLY
000005 //*
000006 //PROCLIB  JCLLIB ORDER=(TEAM??.BPBAR.JCL)
000007 // SET STEPLIB=WMQICPV6
000008 // SET APPLOAD='TEAM??.IP13.LOAD'
000009 // SET MQLOAD1='WMQ800.SCSQLOAD'
000010 // SET MQLOAD2='WMQ800.SCSQAUTH'
000011 // SET MQLOAD3='WMQ800.SCSQLOAD'
000012 // SET THISPDS='TEAM??.BPBAR.JCL'
000013 // SET QM='QML?'
000014 // SET SUMMARY='TEAM??.DOCUMENT.SUMMARY'
000015 // SET DB2='DSNA*'     DB2 NAME USED IN REPORTING CPU
```

   b) Change the 'QML?' to the queue manager you are using.

```
EDIT       TEAMXX.BPBAR.JCL(V8NOIO10) - 01.07                 CHARS 'TEAM??' changed
Command ===> c QML? QML3 all_                                       Scroll ===> CSR
****** ********************************* Top of Data ***********************************
==CHG> //TEAMXXIO JOB (????,????),'BBAR No IO',NOTIFY=????????
000002 //*
000003 //* THE FIRST STEP IS TO LOAD 300 MESSAGES ON A QUEUE THAT SHOULD NOT GET
000004 //* WRITTEN TO THE PAGESET, IN BUFFERPOOL ONLY
000005 //*
==CHG> //PROCLIB  JCLLIB ORDER=(TEAMXX.BPBAR.JCL)
000007 // SET STEPLIB=WMQICPV6
==CHG> // SET APPLOAD='TEAMXX.IP13.LOAD'
000009 // SET MQLOAD1='WMQ800.SCSQLOAD'
000010 // SET MQLOAD2='WMQ800.SCSQAUTH'
000011 // SET MQLOAD3='WMQ800.SCSQLOAD'
==CHG> // SET THISPDS='TEAMXX.BPBAR.JCL'
000013 // SET QM='QML?'
==CHG> // SET SUMMARY='TEAMXX.DOCUMENT.SUMMARY'
```

   c) Save the member by entering 'save' on the command line, r by exiting the member via the F3 key

4) Make the same changes to these other members of the dataset:
   V8NOIO11
   V8YSIO10
   V8YSIO11

5) Each job has four steps

Stepname	Program executed	Purpose
LOADQ	OEMPUTX from IP13	Loads 4K messages onto the specified queue
LOADUSE	CSQUTIL	Issues a DISPLAY USAGE command to show the current Buffer and pageset use
MGETQ	MGET from IP13	Reads the messages from the specified queue
MQGETUSE	CSQUTIL	Issues a DISPLAY USAGE command to show the current Buffer and pageset use

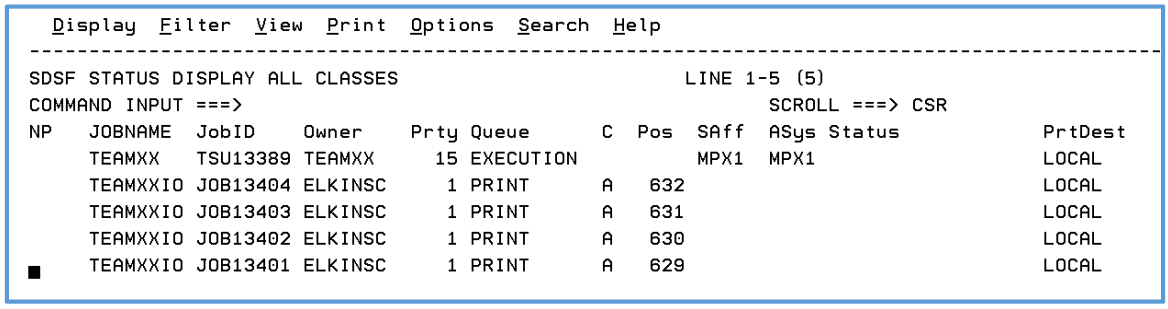
6) Reopen V8NOIO10 and submit the JCL. The JCL does notify when it job is complete, but you may have to hit the enter key a couple of times to get the notification.

```
16.11.19 JOB13401 $HASP165 TEAMXXIO ENDED AT WSC300  MAXCC=0000 CN(INTERNAL)
*** ■
```

7) If the job does not return fairly quickly, chances are good that it has been submitted to run from the wrong LPAR. For example, if the job is referencing an even number queue manager and is submitted to run on MPX1 it will continue to try to access an unknown queue manager for some time. If that is the case, purge the job and resubmit from the correct LPAR.
8) Follow the same steps for submitting the other JCL members, waiting for each job to complete before submitting the last one. This is very important, as the running two jobs against the same resource pool will contaminate the results.

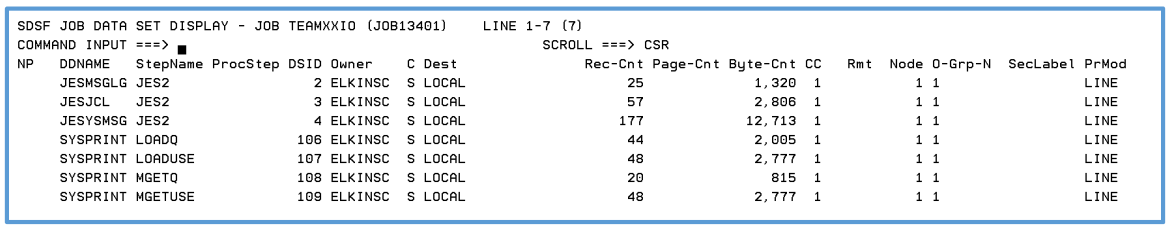
Step 2 – Evaluating the findings

1) One all the jobs have run to completion, navigate the SDSF status display panel. Use =SDSF.ST in the command line of any TSO screen. If you do not see your jobs in the list, the prefix probably needs to be changed. Use the command 'prefix TEAMXX*' where the XX is replaced with your team number.



```
  Display  Filter  View  Print  Options  Search  Help
------------------------------------------------------------------------------------------------
SDSF STATUS DISPLAY ALL CLASSES                          LINE 1-5 (5)
COMMAND INPUT ===>                                       SCROLL ===> CSR
NP   JOBNAME  JobID    Owner    Prty Queue      C  Pos  SAff  ASys Status            PrtDest
     TEAMXX   TSU13389 TEAMXX     15 EXECUTION          MPX1  MPX1                  LOCAL
     TEAMXXIO JOB13404 ELKINSC     1 PRINT      A   632                             LOCAL
     TEAMXXIO JOB13403 ELKINSC     1 PRINT      A   631                             LOCAL
     TEAMXXIO JOB13402 ELKINSC     1 PRINT      A   630                             LOCAL
■    TEAMXXIO JOB13401 ELKINSC     1 PRINT      A   629                             LOCAL
```

2) Use the question mark to expand the first job run, the one with the lowest job ID number.



```
SDSF JOB DATA SET DISPLAY - JOB TEAMXXIO (JOB13401)    LINE 1-7 (7)
COMMAND INPUT ===> ■                                   SCROLL ===> CSR
NP   DDNAME   StepName ProcStep DSID Owner    C Dest          Rec-Cnt Page-Cnt Byte-Cnt CC   Rmt  Node O-Grp-N  SecLabel PrMod
     JESMSGLG JES2                2 ELKINSC  S LOCAL              25         1     1,320  1          1 1                LINE
     JESJCL   JES2                3 ELKINSC  S LOCAL              57         2     2,806  1          1 1                LINE
     JESYSMSG JES2                4 ELKINSC  S LOCAL             177        12    12,713  1          1 1                LINE
     SYSPRINT LOADQ             106 ELKINSC  S LOCAL              44         2     2,005  1          1 1                LINE
     SYSPRINT LOADUSE           107 ELKINSC  S LOCAL              48         2     2,777  1          1 1                LINE
     SYSPRINT MGETQ             108 ELKINSC  S LOCAL              20         1       815  1          1 1                LINE
     SYSPRINT MGETUSE           109 ELKINSC  S LOCAL              48         2     2,777  1          1 1                LINE
```

3) Select the LOADQ SYSPRINT file. If unfamiliar with the output of OEMPUTX, please read the documentation on the test job from the IP13 SupportPac.

4) Page down until the 'Total Transactions' thru 'Avg App CPU' is shown.

```
Total Transactions  : 300
Elapsed Time        :    0.016 seconds
Application CPU Time:    0.012 seconds  (76.8%)
Transaction Rate    : 19244.340 trans/sec
-----------------------------------------------------
Round trip per msg  :       51 microseconds
Avg App CPU per msg :       39 microseconds
-----------------------------------------------------
```

Make note of the following information from the test:

Total Transactions: _____
Transaction Rate: _____
Round trip per msg: _____
Avg App CPU per msg:_____

5) Return to the output list and select the SYSPRINT from the LOADUSE step. This is the output from the 'DISPLAY USAGE' command.

```
CSQI010I QML3 Page set usage ...
  Page Buffer     Total    Unused  Persistent  NonPersist Expansion
   set   pool     pages     pages  data pages  data pages        count
_    0      0      1078      1039          39           0 USER       0
_    1      0      1078      1046          32           0 USER       0
_    2      1      1078      1076           2           0 USER       0
_    3      2      1078      1077           0           1 USER       0
_    4      3      1078      1066          12           0 USER       0
_   10     10      2698      2698           0           0 USER       0
_   11     11      2698      2698           0           0 USER       0
_   12     12      1078      1078           0           0 USER       0
_   13     13      1078      1078           0           0 USER       0
_   14     14      1078      1078           0           0 USER       0
_   15     15      1078      1078           0           0 USER       0
_   16     16      1078      1078           0           0 USER       0
_   17     17      1078      1078           0           0 USER       0
_   18     18      1078      1078           0           0 USER       0
_   19     19      1078      1078           0           0 USER       0
_   20     20      1078       473           0         605 USER       0
_   21     21      1078      1078           0           0 USER       0
 End of page set report
CSQI065I QML3 Buffer pool attributes ...
  Buffer  Available  Stealable   Stealable  Page      Location
    pool    buffers    buffers  percentage  class
_      0      50000      49951          99  4KB       BELOW
_      1      20000      19999          99  4KB       BELOW
_      2      50000      49994          99  4KB       ABOVE
_      3      20000      19995          99  4KB       BELOW
_     10       1000        999          99  4KB       BELOW
_     11       1000        999          99  4KB       ABOVE
_     12       1000        999          99  4KB       BELOW
_     13       1000        999          99  4KB       ABOVE
_     14       1000        999          99  4KB       BELOW
_     15       1000        999          99  4KB       ABOVE
_     16       1000        999          99  4KB       BELOW
_     17       1000        999          99  4KB       ABOVE
_     18       1000        999          99  4KB       BELOW
_     19       1000        999          99  4KB       ABOVE
_     20       1000        393          39  4KB       BELOW
_     21       1000        999          99  4KB       ABOVE
```

What is the value of Unused pages in the pageset your team is using?

_____

Does it differ from the sample given (using pageset 20)? If so, what might be the reason?

_____

How many stealable buffers remain in the bufferpool being used for this test after this job has run? (using bufferpool 20)

_____

6) Return to the output list and select the SYSPRINT from the MGETQ step. This is the output step that reads the messages from the queue.

```
About to get 99999999 messages from:
  Qname   = TEAMXX.BBAR.QUEUE
  Qmgr    = QML3
Wait interval is 10 seconds
Quiet mode - Messages will not be printed
Buffer size is 1000 bytes
--------------------------------------------------------
Starting at 2014-03-26 02:42:08.611071

--------------------------------------------------------
Total Messages      : 300
Elapsed Time        : 0.008013 seconds
Message Rate        : 37439.11 msgs/sec
Average MQGET Time  : 0.033360 seconds
--------------------------------------------------------
Application CPU Time: 0.007600 seconds  (94.8%)
CPU per Message     : 0.0253 milliseconds
```

Make note of the following information from the test:

Total Messages: _____
Message Rate: _____
CPU per msg:_____

7) Return to the list of completed jobs, and expand the next one that ran. It should be the test for the above the bar bufferpool with no pageset I/O. Slecting the LOADQ SYSPRINT output the queue name should be TEAMXX.ABAR.QUEUE, where the XX is your team number.

8) Page down until the 'Total Transactions' thru 'Avg App CPU' is shown.

```
Total Transactions  : 300
Elapsed Time        :    0.016 seconds
Application CPU Time:    0.012 seconds  (76.4%)
Transaction Rate    : 19174.180 trans/sec
-----------------------------------------------------
Round trip per msg  :       52 microseconds
Avg App CPU per msg :       39 microseconds
-----------------------------------------------------
```

Make note of the following information from the test:

Total Transactions: _____
Transaction Rate: _____
Round trip per msg: _____
Avg App CPU per msg:_____

9) Compare the numbers with those from the below the bar test. The sample test captured the following:

```
Total Transactions  : 300
Elapsed Time        :    0.016 seconds
Application CPU Time:    0.012 seconds  (76.8%)
Transaction Rate    : 19244.340 trans/sec
-----------------------------------------------------
Round trip per msg  :       51 microseconds
Avg App CPU per msg :       39 microseconds
-----------------------------------------------------
```

Note that the difference observed by the OEMPUTX process is about 1 microsecond in the roundtrip time, and slightly lower transaction rate. The average CPU consumption was the same.

10) Return to the output list and select the SYSPRINT from the LOADUSE step. This is the output from the 'DISPLAY USAGE' command.

```
CSQI010I QML3 Page set usage ...
   Page Buffer     Total    Unused  Persistent  NonPersist Expansion
    set   pool     pages     pages  data pages  data pages        count
_     0      0      1078      1039          39           0 USER       0
_     1      0      1078      1046          32           0 USER       0
_     2      1      1078      1076           2           0 USER       0
_     3      2      1078      1077           0           1 USER       0
_     4      3      1078      1066          12           0 USER       0
_    10     10      2698      2698           0           0 USER       0
_    11     11      2698      2698           0           0 USER       0
_    12     12      1078      1078           0           0 USER       0
_    13     13      1078      1078           0           0 USER       0
_    14     14      1078      1078           0           0 USER       0
_    15     15      1078      1078           0           0 USER       0
_    16     16      1078      1078           0           0 USER       0
_    17     17      1078      1078           0           0 USER       0
_    18     18      1078      1078           0           0 USER       0
_    19     19      1078      1078           0           0 USER       0
_    20     20      1078      1078           0           0 USER       0
_    21     21      1078       473           0         605 USER       0
 End of page set report
CSQI065I QML3 Buffer pool attributes ...
   Buffer  Available  Stealable   Stealable  Page      Location
     pool    buffers    buffers  percentage  class
_       0      50000      49951          99  4KB       BELOW
_       1      20000      19999          99  4KB       BELOW
_       2      50000      49994          99  4KB       ABOVE
_       3      20000      19953          99  4KB       BELOW
_      10       1000        999          99  4KB       BELOW
_      11       1000        999          99  4KB       ABOVE
_      12       1000        999          99  4KB       BELOW
_      13       1000        999          99  4KB       ABOVE
_      14       1000        999          99  4KB       BELOW
_      15       1000        999          99  4KB       ABOVE
_      16       1000        999          99  4KB       BELOW
_      17       1000        999          99  4KB       ABOVE
_      18       1000        999          99  4KB       BELOW
_      19       1000        999          99  4KB       ABOVE
_      20       1000        393          39  4KB       BELOW
_      21       1000        393          39  4KB       ABOVE
```

What is the value of Unused pages in the pageset your team is using?

_____

Does it differ from the sample given (using pageset 21)? If so, what might be the reason?

_____

Does the number of Unused pages in the job you ran differ from the below the bar test?

How many stealable buffers remain in the bufferpool being used for this test after this job has run? (using bufferpool 21)

_____

Does the number of stealable buffers in the job you ran differ from the below the bar test?

11) Return to the output list and select the SYSPRINT from the MGETQ step. This is the output step that reads the messages from the queue.

```
About to get 99999999 messages from:
  Qname   = TEAMXX.ABAR.QUEUE
  Qmgr    = QML3
Wait interval is 10 seconds
Quiet mode - Messages will not be printed
Buffer size is 1000 bytes
--------------------------------------------------------
Starting at 2014-03-26 02:42:59.714637

--------------------------------------------------------
Total Messages      : 300
Elapsed Time        : 0.007851 seconds
Message Rate        : 38211.67 msgs/sec
Average MQGET Time  : 0.033360 seconds
--------------------------------------------------------
Application CPU Time: 0.007400 seconds  (94.3%)
CPU per Message     : 0.0247 milliseconds
```

Make note of the following information from the test:

Total Messages: _____
Message Rate: _____
CPU per msg:_____

12) Compare the MGET result with the below the bar results. The sample given was:

```
About to get 99999999 messages from:
  Qname   = TEAMXX.BBAR.QUEUE
  Qmgr    = QML3
Wait interval is 10 seconds
Quiet mode - Messages will not be printed
Buffer size is 1000 bytes
--------------------------------------------------------
Starting at 2014-03-26 02:42:08.611071

--------------------------------------------------------
Total Messages      : 300
Elapsed Time        : 0.008013 seconds
Message Rate        : 37439.11 msgs/sec
Average MQGET Time  : 0.033360 seconds
--------------------------------------------------------
Application CPU Time: 0.007600 seconds  (94.8%)
CPU per Message     : 0.0253 milliseconds
```

How does the Message Rate vary from the below the bar test you ran:

_____

And the CPU per msg rate?:_____

Interestingly enough, in the sample test the get process from the above the bar bufferpool the performance was actually better.

TECHTIP 8: Your mileage will vary – it is critical that customers test this for themselves in a production like environment.

13) Return to the list of run jobs and select the next one, below the bar with I/O.
14) Again, examine the output from the LOADQ step. The results from the sample test look as follows:

```
Total Transactions  : 1200
Elapsed Time        :    0.974 seconds
Application CPU Time:    0.050 seconds  (5.1%)
Transaction Rate    : 1232.367 trans/sec
-----------------------------------------------------
Round trip per msg  :      811 microseconds
Avg App CPU per msg :       41 microseconds
```

In this test there should be I/O to the pageset. Compare the transaction rate, roundtrip and average CPU between this test and the below the bar BP with no I/O. In the samples the comparison would looks as follows:

Test Type	Transaction Rate	Roundtrip	Average CPU
BP below, no IO	19.244.340	51	39
BP below, IO	1232.367	811	41

Were your results similar?

15) Return to the output list, and examine the display usage results following the LOADQ (called the LOADUSE) step. Were there differences in the usage shown for either pageset or bufferpool from the earlier tests?

_____

16) Return to the output list and examine the MGETQ output. The sample output shows the following:

```
About to get 99999999 messages from:
  Qname   = TEAMXX.BBAR.QUEUE
  Qmgr    = QML3
Wait interval is 10 seconds
Quiet mode - Messages will not be printed
Buffer size is 1000 bytes
---------------------------------------------------------
Starting at 2014-03-26 02:44:56.799397

---------------------------------------------------------
Total Messages      : 1200
Elapsed Time        : 0.367602 seconds
Message Rate        : 3264.40 msgs/sec
Average MQGET Time  : 0.008640 seconds
---------------------------------------------------------
Application CPU Time: 0.052700 seconds  (14.3%)
CPU per Message     : 0.0439 milliseconds
```

Compare the message rate and CPU per message values. In the samples test, we observed the following:

Test Type	Transaction Rate	Average MQGET time	Average CPU
BP below, no IO	37439.11	.0076	.0253
BP below, IO	3264.40	.008640	.0439 milliseconds

TECHTIP 9: I/O can be expensive. The costs, both CPU and responsiveness, are very dependent on the underlying hardware and software that drives the I/O. I/O cannot be avoided on persistent messages, as those must be written to the logs, but it can be for non-persistent messages.

17) Return to the list of completed jobs, and expand the next one that ran. It should be the test for the above the bar bufferpool with pageset I/O. Selecting the LOADQ SYSPRINT output the queue name should be TEAMXX.ABAR.QUEUE, where the XX is your team number.

18) Page down until the 'Total Transactions' thru 'Avg App CPU' is shown.

```
Total Transactions  : 1200
Elapsed Time        :    0.972 seconds
Application CPU Time:    0.050 seconds  (5.1%)
Transaction Rate    : 1234.466 trans/sec
-----------------------------------------------------
Round trip per msg  :      810 microseconds
Avg App CPU per msg :       41 microseconds
-----------------------------------------------------
```

Compare the rates between the above the bar wit and without I/O. The same tests showed the following.

Test Type	Transaction Rate	Roundtrip	Average CPU
BP above, no IO	19174.180	52	39
BP above, IO	1234.466	810	41

19) Return to the output list, and examine the display usage results following the LOADQ (called the LOADUSE) step. Were there differences in the usage shown for either pageset or bufferpool?

_____

20) Return to the output list and examine the MGETQ output. The sample output shows the following:

```
About to get 99999999 messages from:
  Qname   = TEAMXX.ABAR.QUEUE
  Qmgr    = QML3
Wait interval is 10 seconds
Quiet mode - Messages will not be printed
Buffer size is 1000 bytes
--------------------------------------------------------
Starting at 2014-03-26 02:50:41.562665

--------------------------------------------------------
Total Messages      : 1200
Elapsed Time        : 0.354260 seconds
Message Rate        : 3387.34 msgs/sec
Average MQGET Time  : 0.008629 seconds
--------------------------------------------------------
Application CPU Time: 0.050400 seconds  (14.2%)
CPU per Message     : 0.0420 milliseconds
--------------------------------------------------------
```

Compare the message rate and CPU per message values. In the samples test, we observed the following:

Test Type	Transaction Rate	Average MQGET time	Average CPU
BP above, no IO	38211.67	.0074	.0247
BP above, IO	3387.34	.050400	.042 milliseconds

21)
22)
23)
24)

Conclusion

These simple tests illustrate that the runtime costs of using the WMQ V8 feature of buffers above the bar are not significantly higher than the buffers below the bar in a limited test environment. However, this may not always be the case. If there is not enough real memory to support the above the bar buffers, z/OS paging will occur if he buffer pages are not fixed. If the buffer pages are fixed, the memory requirements could impact other application performance.

Customers should evaluate the costs in their environment to avoid ‘surprises’.

Please also note we had hoped to present the new SMF bufferpool data , but the MP16 print program form the beta downloads is not formatting them as of the date this was written. That has been reported as a problem to the lab, and we hope to have a resolution soon.