Setting up shared message datasets for message offloading

Audience level: knowledge of MQ or z/OS
Skillset: z/OS Systems Programming, MQ Administration

Background:

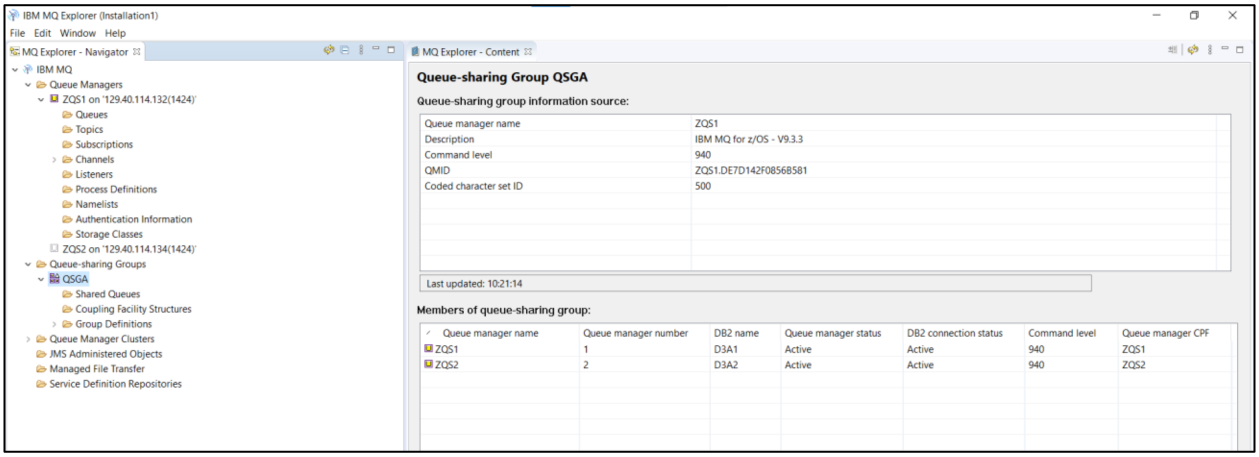
SMDS are the preferred method for offloading large messages in queue-sharing groups. By default, large messages will be offloaded to Db2 blobs. In this lab, we will compare the differences in both transaction rate and CPU consumption of SMDS versus blobs.

Overview of exercise:

1. Customize and run CSQ4SMDS in SCSQPROC
2. Alter the CF Structure properties
3. Run display commands to verify the SMDS configuration is correct

Steps of exercise:

1. Before getting started, verify that the below configuration is in place by viewing MQ Explorer. You should see connections to ZQS1, ZQS2, and you should see a QSGA Queue-sharing group visible.



Queue-sharing Group QSGA

Queue-sharing group information source:

Queue manager name	ZQS1
Description	IBM MQ for z/OS - V9.3.3
Command level	940
QMID	ZQS1.DE7D142F0856B581
Coded character set ID	500

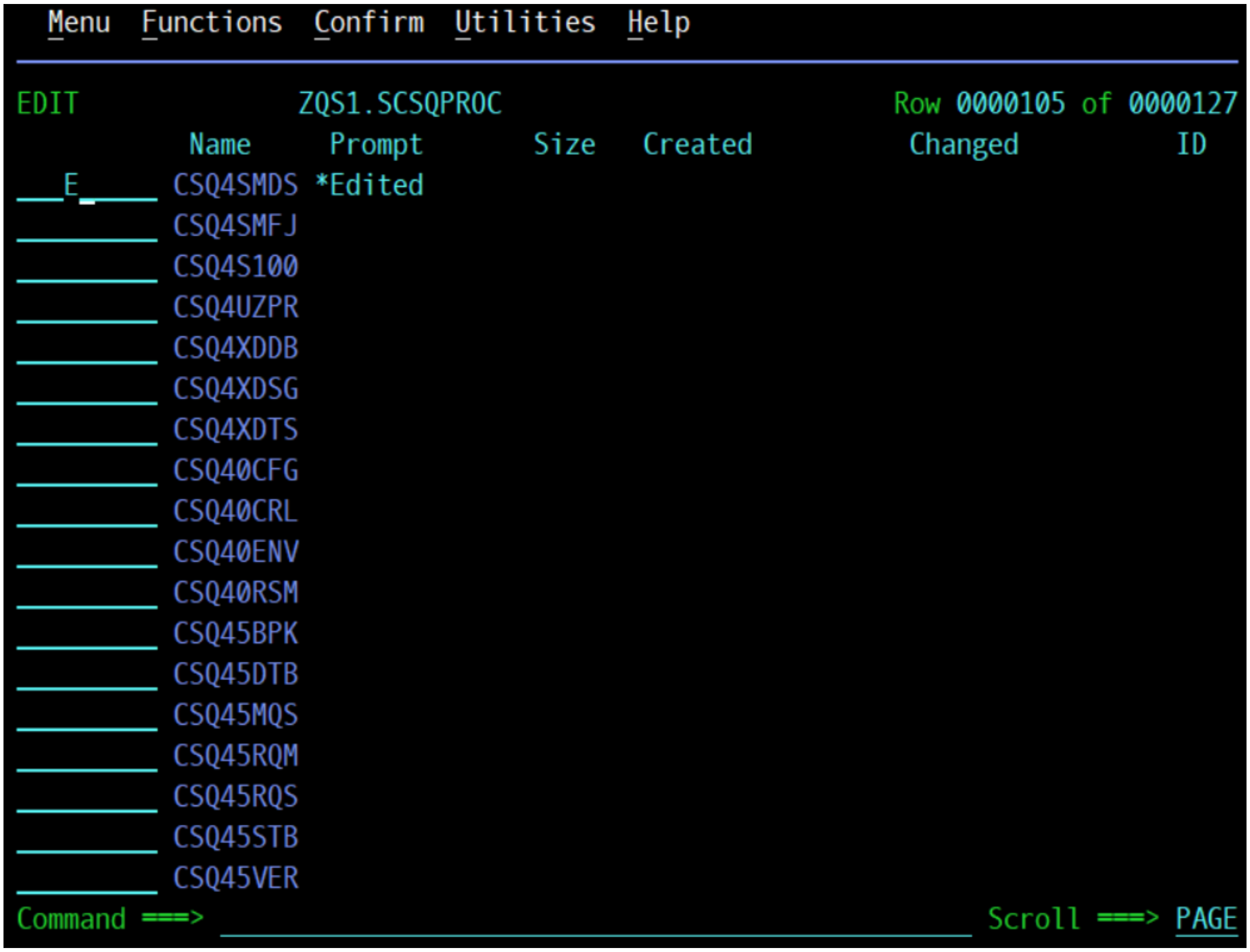
Last updated: 10:21:14

Members of queue-sharing group:

Queue manager name	Queue manager number	DB2 name	Queue manager status	DB2 connection status	Command level	Queue manager CPF
ZQS1	1	D3A1	Active	Active	940	ZQS1
ZQS2	2	D3A2	Active	Active	940	ZQS2

2. On MQS1, navigate to ZQS1.SCSQPROC from the ISPF main menu by using option 3.4.

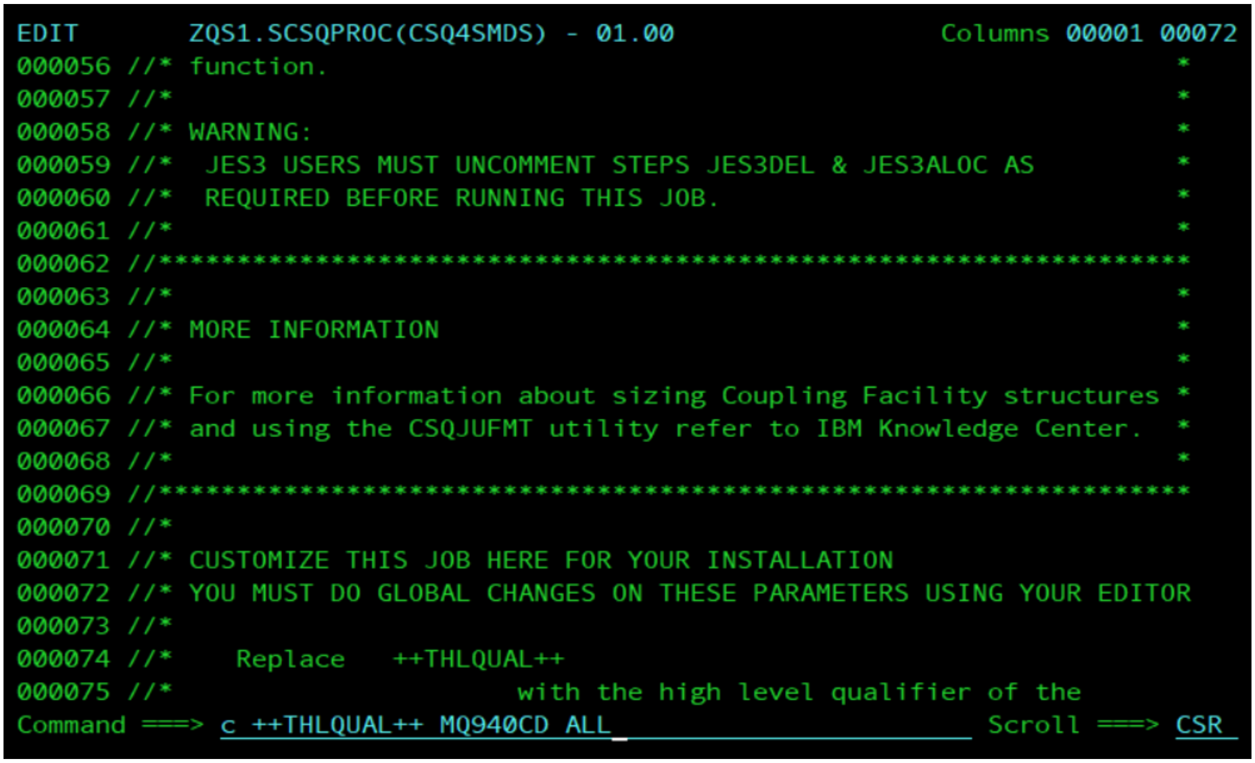
3. Using the F7 and F8 keys to navigate up and down, find member CSQ4SMDS. Type ‘e’ next to the member CSQ4SMDS to edit the contents.



```
  Menu  Functions  Confirm  Utilities  Help
EDIT              ZQS1.SCSQPROC                       Row 0000105 of 0000127
           Name     Prompt       Size   Created           Changed          ID
___E_____ CSQ4SMDS *Edited
_________ CSQ4SMFJ
_________ CSQ4S100
_________ CSQ4UZPR
_________ CSQ4XDDB
_________ CSQ4XDSG
_________ CSQ4XDTS
_________ CSQ40CFG
_________ CSQ40CRL
_________ CSQ40ENV
_________ CSQ40RSM
_________ CSQ45BPK
_________ CSQ45DTB
_________ CSQ45MQS
_________ CSQ45RQM
_________ CSQ45RQS
_________ CSQ45STB
_________ CSQ45VER
Command ===> ____________________________________________  Scroll ===> PAGE
```

Looking through CSQ4SMDS, you will notice it using the IDCAMS utility to build a VSAM data set. The VSAM data set it builds will act as message storage for messages using SMDS. Each queue manager in the QSG will need its own VSAM data set for a SMDS configuration.

4. CSQ4SMDS asks us to customize the below variables. You will use the format 'c ++THLQUAL++ MQ940CD ALL' in the command line to adjust all of these variables

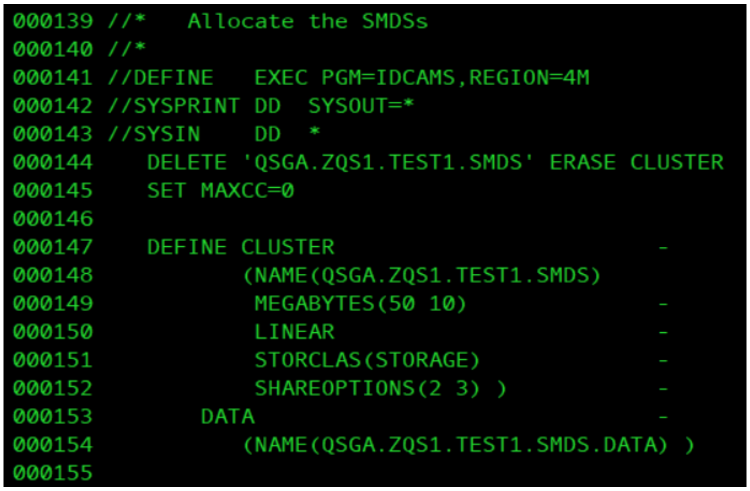


```
EDIT       ZQS1.SCSQPROC(CSQ4SMDS) - 01.00                  Columns 00001 00072
000056 //* function.                                                          *
000057 //*                                                                    *
000058 //* WARNING:                                                           *
000059 //*  JES3 USERS MUST UNCOMMENT STEPS JES3DEL & JES3ALOC AS             *
000060 //*  REQUIRED BEFORE RUNNING THIS JOB.                                 *
000061 //*                                                                    *
000062 //*********************************************************************
000063 //*                                                                    *
000064 //* MORE INFORMATION                                                   *
000065 //*                                                                    *
000066 //* For more information about sizing Coupling Facility structures *
000067 //* and using the CSQJUFMT utility refer to IBM Knowledge Center.  *
000068 //*                                                                    *
000069 //*********************************************************************
000070 //*
000071 //* CUSTOMIZE THIS JOB HERE FOR YOUR INSTALLATION
000072 //* YOU MUST DO GLOBAL CHANGES ON THESE PARAMETERS USING YOUR EDITOR
000073 //*
000074 //*    Replace   ++THLQUAL++
000075 //*                       with the high level qualifier of the
Command ===> c ++THLQUAL++ MQ940CD ALL                         Scroll ===> CSR
```

++THLQUAL++	**→**	**MQ940CD**
++HLQ++	**→**	**QSGA**
++QMGR++	**→**	**ZQS1**
++CFSTRUCT++	**→**	**TEST1**
++PRI++	**→**	**50**
++SEC++	**→**	**10**
++LANGLETTER++	**→**	**E**
DATACLAS(EXTENDED)	**→**	**STORCLAS(STORAGE)**

5. Your CSQ4SMDS should reflect the ‘after’. At this point, you can go ahead and submit the job by typing ‘submit’ on the command line and hitting enter.

After changes:



```
000139 //*   Allocate the SMDSs
000140 //*
000141 //DEFINE   EXEC PGM=IDCAMS,REGION=4M
000142 //SYSPRINT DD  SYSOUT=*
000143 //SYSIN    DD  *
000144     DELETE 'QSGA.ZQS1.TEST1.SMDS' ERASE CLUSTER
000145     SET MAXCC=0
000146
000147     DEFINE CLUSTER                                -
000148            (NAME(QSGA.ZQS1.TEST1.SMDS)
000149             MEGABYTES(50 10)                      -
000150             LINEAR                                -
000151             STORCLAS(STORAGE)                     -
000152             SHAREOPTIONS(2 3) )                   -
000153         DATA                                      -
000154            (NAME(QSGA.ZQS1.TEST1.SMDS.DATA) )
000155
```

6. Submit the job using the command 'submit' and pressing enter.

7. Now, turn over to the MQS2 image. Here, log in and navigate to ZQS2.SCSQPROC using option 3.4 from the ISPF main menu.

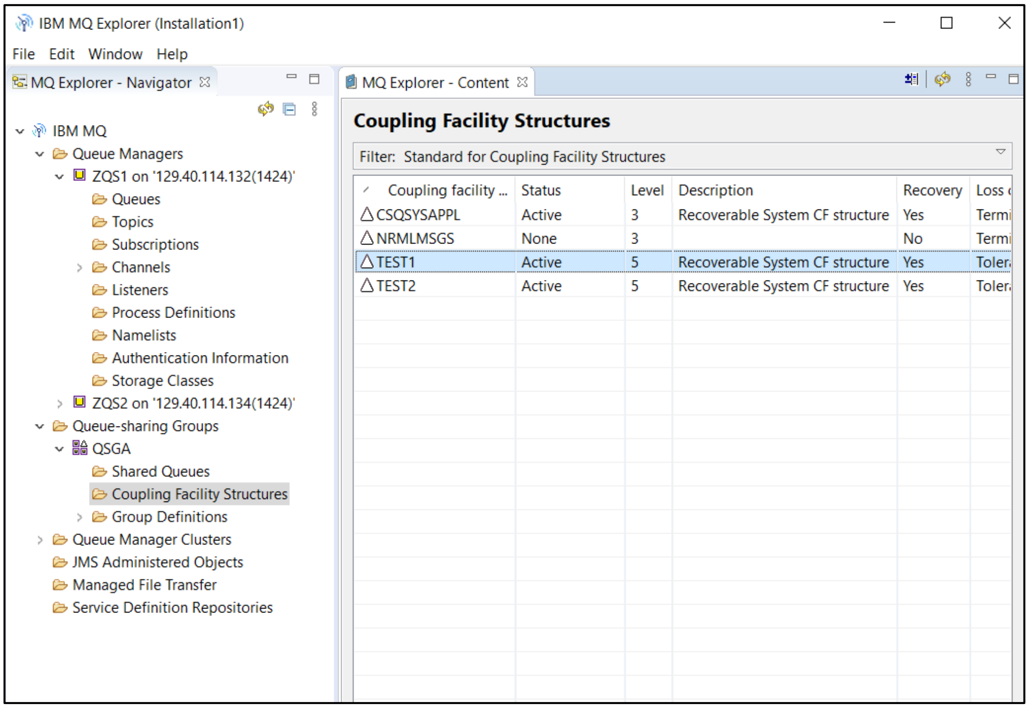
8. We are going to execute the CSQ4SMDS JCL for our other queue manager, ZQS2.

9. Make the same changes you made to ZQS2.SCSQPROC(CSQ4SMDS) in step 4.

10. Submit the job using the command 'submit' and pressing enter.

11. Now, we have the shared message data sets set up. We need to tie them to our coupling facility structures.

12. On MQ Explorer, navigate to the CF structure screen like below:



Coupling Facility Structures

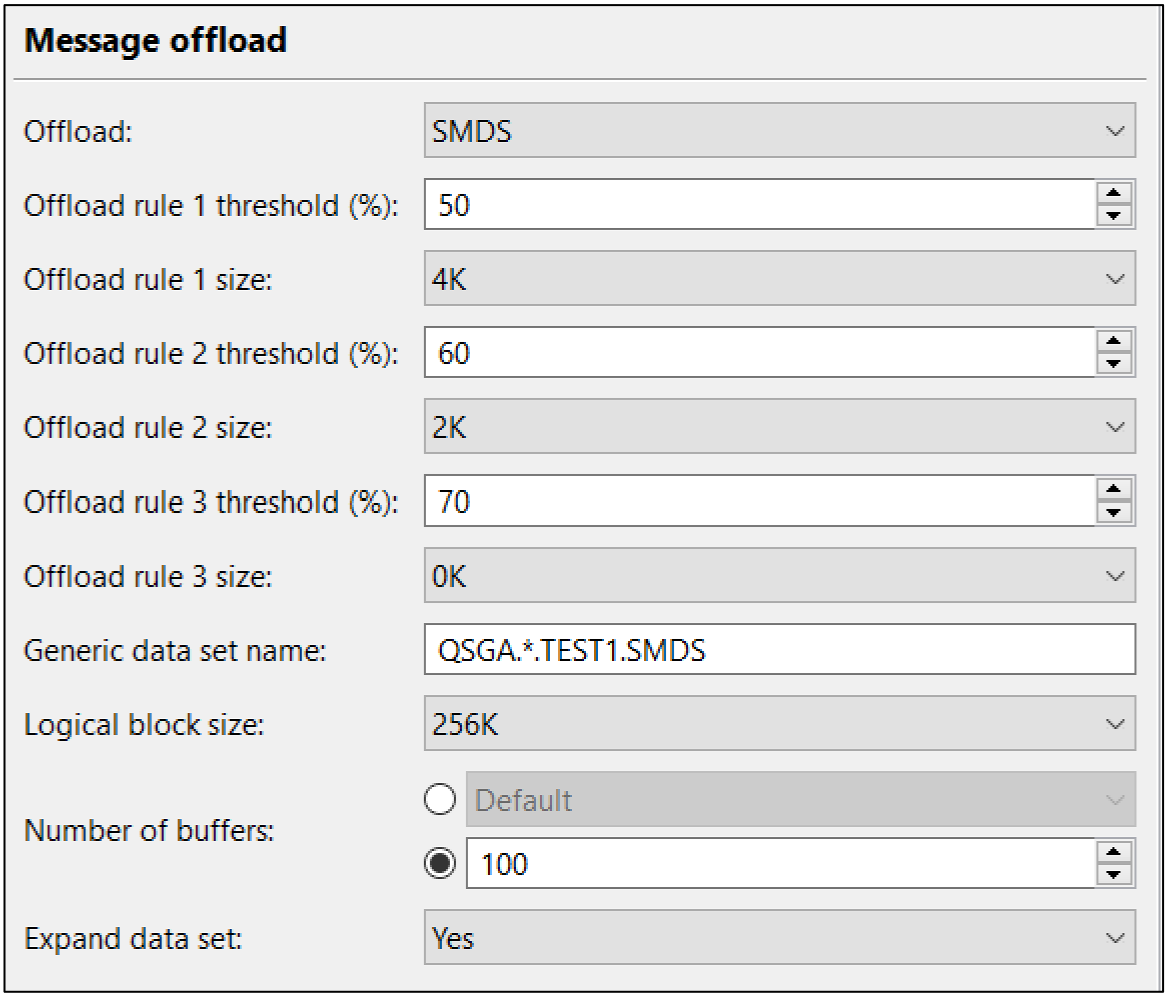
Filter: Standard for Coupling Facility Structures

Coupling facility ...	Status	Level	Description	Recovery	Loss c
△CSQSYSAPPL	Active	3	Recoverable System CF structure	Yes	Termi
△NRMLMSGS	None	3		No	Termi
△TEST1	Active	5	Recoverable System CF structure	Yes	Toler
△TEST2	Active	5	Recoverable System CF structure	Yes	Toler

13. Right click on TEST1, and select the last option 'Properties…'.

14. In the 'Properties' window, select 'Message Offload'.

15. Adjust the ‘Offload’ and ‘Generic data set name’ fields to reflect the screen below:

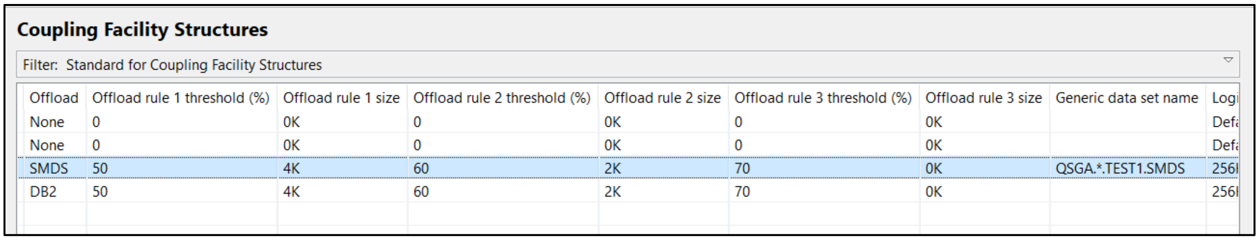


Message offload

Offload:	SMDS
Offload rule 1 threshold (%):	50
Offload rule 1 size:	4K
Offload rule 2 threshold (%):	60
Offload rule 2 size:	2K
Offload rule 3 threshold (%):	70
Offload rule 3 size:	0K
Generic data set name:	QSGA.*.TEST1.SMDS
Logical block size:	256K
Number of buffers:	○ Default ● 100
Expand data set:	Yes

Note: Using an * here means that TEST1 can take effect for both of the ZQS1 and ZQS2 shared message data sets we built.

16. Press ‘Apply’. The updates should now be visible in the CF Structures screen.

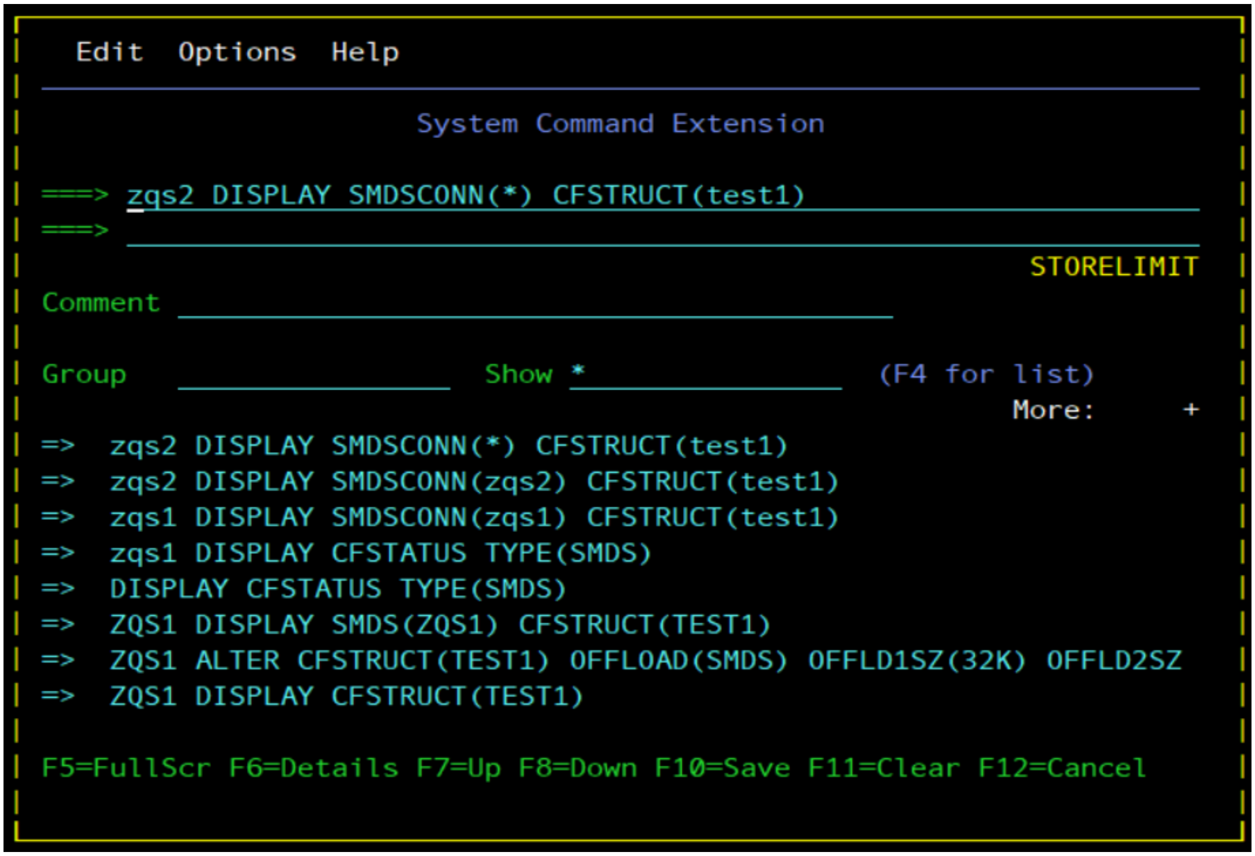


Coupling Facility Structures

Filter: Standard for Coupling Facility Structures

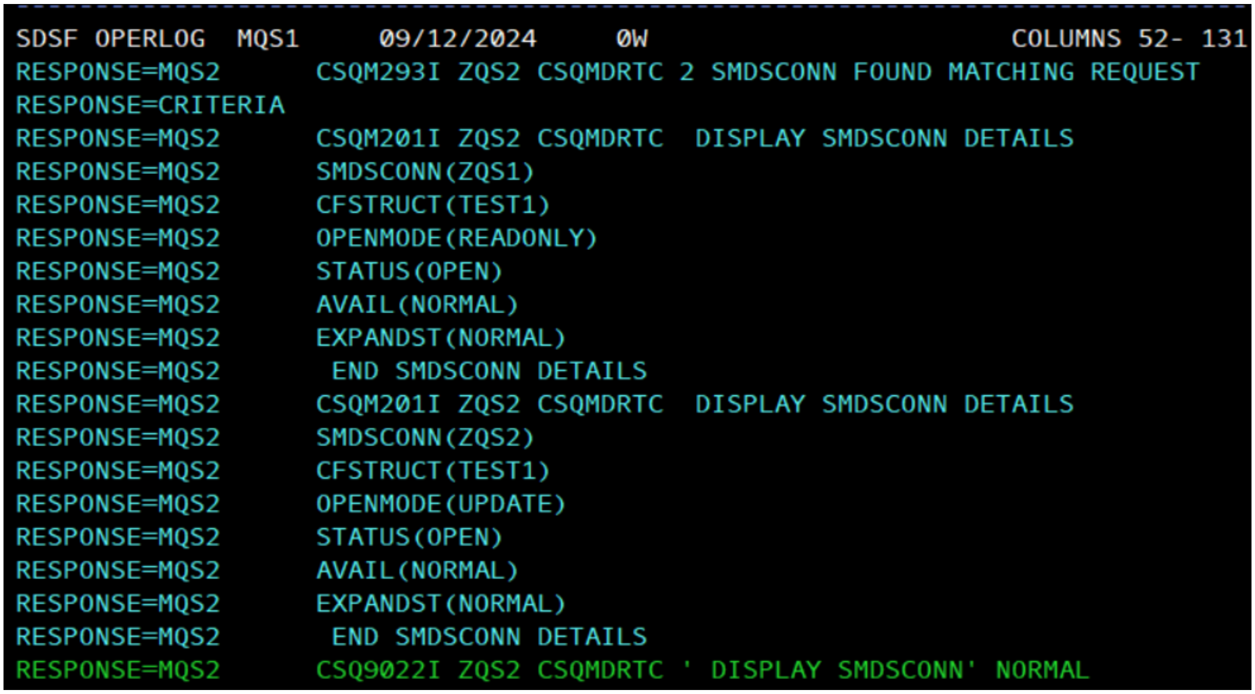
Offload	Offload rule 1 threshold (%)	Offload rule 1 size	Offload rule 2 threshold (%)	Offload rule 2 size	Offload rule 3 threshold (%)	Offload rule 3 size	Generic data set name	Logi
None	0	0K	0	0K	0	0K		Defa
None	0	0K	0	0K	0	0K		Defa
SMDS	50	4K	60	2K	70	0K	QSGA.*.TEST1.SMDS	256I
DB2	50	4K	60	2K	70	0K		256I

17. Congratulations! You have now configured SMDS on your z/OS environment. Let’s verify from SDSF on MQS1. From the command window, enter the following command:



```
   Edit  Options  Help

                     System Command Extension

===> zqs2 DISPLAY SMDSCONN(*) CFSTRUCT(test1)
===>
                                                          STORELIMIT
Comment

Group                   Show *                  (F4 for list)
                                                      More:     +
=>  zqs2 DISPLAY SMDSCONN(*) CFSTRUCT(test1)
=>  zqs2 DISPLAY SMDSCONN(zqs2) CFSTRUCT(test1)
=>  zqs1 DISPLAY SMDSCONN(zqs1) CFSTRUCT(test1)
=>  zqs1 DISPLAY CFSTATUS TYPE(SMDS)
=>  DISPLAY CFSTATUS TYPE(SMDS)
=>  ZQS1 DISPLAY SMDS(ZQS1) CFSTRUCT(TEST1)
=>  ZQS1 ALTER CFSTRUCT(TEST1) OFFLOAD(SMDS) OFFLD1SZ(32K) OFFLD2SZ
=>  ZQS1 DISPLAY CFSTRUCT(TEST1)

F5=FullScr F6=Details F7=Up F8=Down F10=Save F11=Clear F12=Cancel
```

Your result should look like the following, no matter whether you use the +cpf prefix zqs1 or zqs2:



```
SDSF OPERLOG  MQS1      09/12/2024     0W                    COLUMNS 52- 131
RESPONSE=MQS2       CSQM293I ZQS2 CSQMDRTC 2 SMDSCONN FOUND MATCHING REQUEST
RESPONSE=CRITERIA
RESPONSE=MQS2       CSQM201I ZQS2 CSQMDRTC  DISPLAY SMDSCONN DETAILS
RESPONSE=MQS2       SMDSCONN(ZQS1)
RESPONSE=MQS2       CFSTRUCT(TEST1)
RESPONSE=MQS2       OPENMODE(READONLY)
RESPONSE=MQS2       STATUS(OPEN)
RESPONSE=MQS2       AVAIL(NORMAL)
RESPONSE=MQS2       EXPANDST(NORMAL)
RESPONSE=MQS2        END SMDSCONN DETAILS
RESPONSE=MQS2       CSQM201I ZQS2 CSQMDRTC  DISPLAY SMDSCONN DETAILS
RESPONSE=MQS2       SMDSCONN(ZQS2)
RESPONSE=MQS2       CFSTRUCT(TEST1)
RESPONSE=MQS2       OPENMODE(UPDATE)
RESPONSE=MQS2       STATUS(OPEN)
RESPONSE=MQS2       AVAIL(NORMAL)
RESPONSE=MQS2       EXPANDST(NORMAL)
RESPONSE=MQS2        END SMDSCONN DETAILS
RESPONSE=MQS2       CSQ9022I ZQS2 CSQMDRTC ' DISPLAY SMDSCONN' NORMAL
```