Title: Setting and using the IBM MQ Console on z/OS

Subtitle: Learn how to view and edit MQ objects using the IBM MQ Console on z/OS

Author: Dorothy Quincy (Dorothy.Quincy@ibm.com)

Estimated time: This tutorial should take about 30 minutes

In this tutorial, you'll learn how to set up the MQ web console on z/OS using basic authentication. You'll learn how to access the MQ console and view and edits MQ objects using it.

The security configuration defined in this tutorial is not sufficient for use in a production environment. For more information regarding MQ web console security, see the linked documentation

The following restrictions apply when using the IBM MQ Console to manage queue managers on z/OS, but with each new version, there is more capability:

- A web console must be created for each new version of MQ on z/OS
- Queue managers on z/OS cannot be created, deleted, started, or stopped.
- Channel initiators on z/OS cannot be started or stopped.

The JCL used in this exercise was originally developed by Mitch Johnson in the IBM Washington Systems Center.

Prerequisites

-Java installed on z/OS, linked here

-IBM MQ installed on z/OS. We are using IBM MQ 9.4 here. See link for more details

-Unix Systems Services active on z/OS. Plan with a system administrator if USS configuration is needed.

-ZFS file system mounted (in our case, we mounted M940CD.SCSQZFS on /usr/lpp/mqm/V9R4MX)

Steps

To configure the console, there are 4 main steps:

-Run the CRTMQWEB executable. This script creates the WebSphere® Liberty user directory that contains the mqm server configuration and log files.

-Modify the XML of the server created by running the CRTMQWEB executable

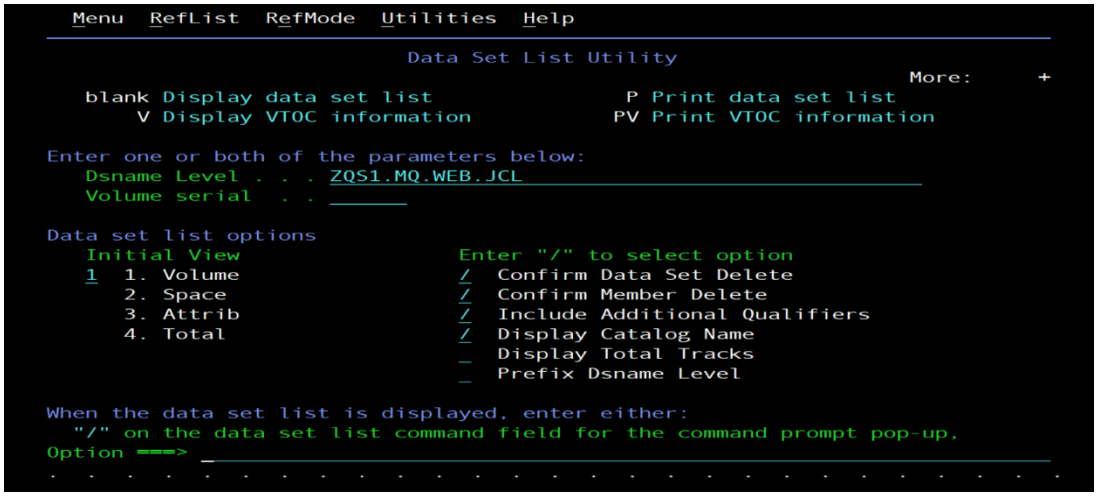
-Add the MQWEBS JCL to the started task library

-Explore the MQ web console functionality

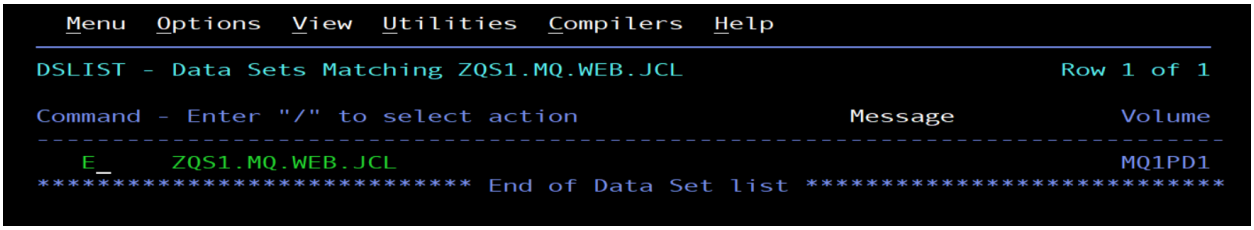
Step 1. Modify the JCL for the CRTMQWEB executable

For this tutorial, I created a data set to house all the relevant MQ Web Console JCL. It is called ZQS1.MQ.WEB.JCL. We will use the MQWEBS, MQANGEL, and CRTMQWEB JCL samples that are contained in ZQS1.MQ.WEB.JCL

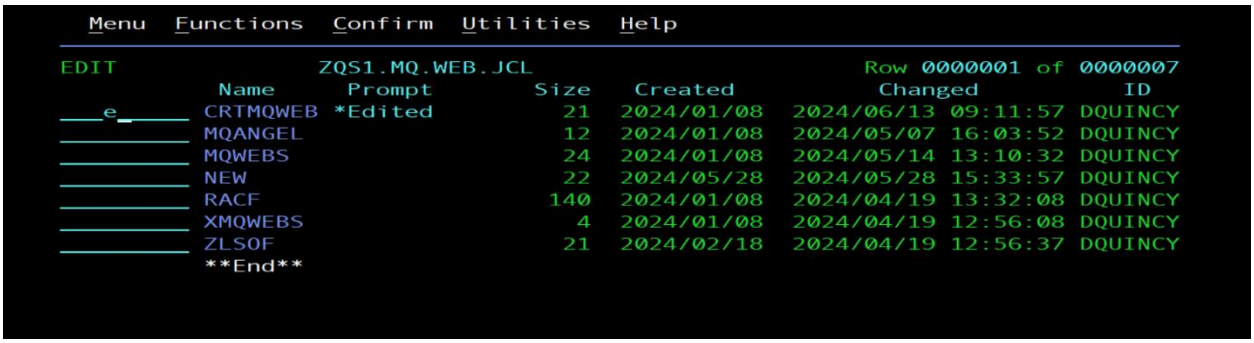
1. The JCL executes the crtmqweb executable in the MQ installation. From the ISPF main menu, go to option 3.4 to search for the ZQS1.MQ.WEB.JCL JCL. In the **Dsname Level** field, specify ZQS1.MQ.WEB.JCL, and press **Enter**.



2. Specify an “e” to the left of the ZQS1.MQ.WEB.JCL data set name to browse its contents in edit-mode.



3. Place an “e” next to CRTMQWEB and press **Enter**. This will put you in edit-mode for the member.



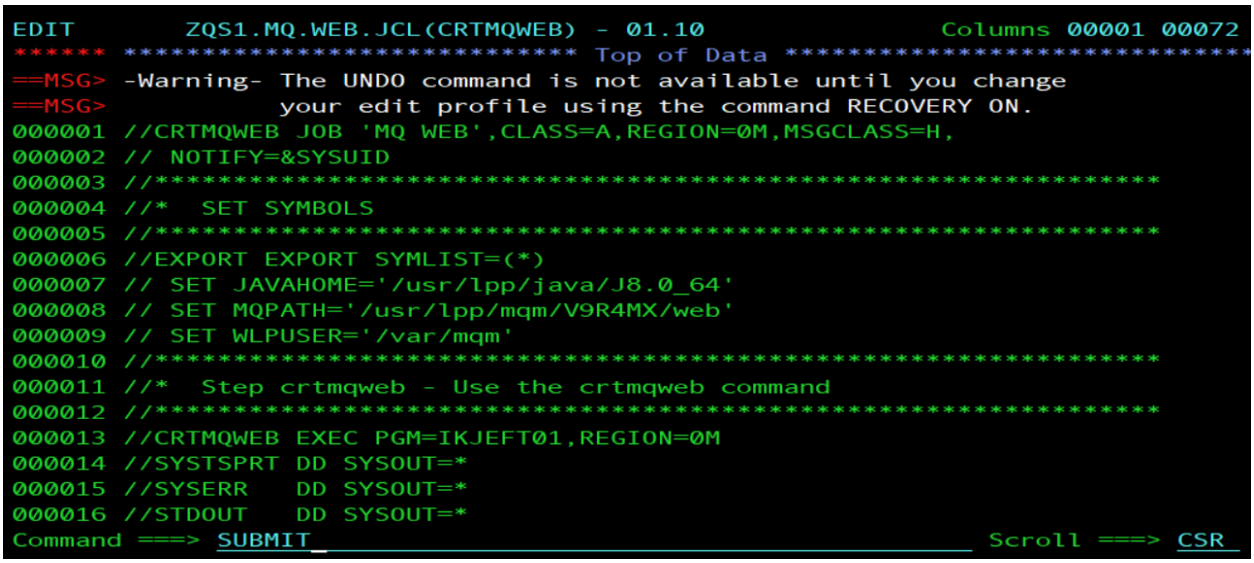
4. In the CRTMQWEB JCL, update the JAVAHOME, MQPATH, and WLPUSER environment variables for your own z/OS environment.

    - JAVAHOME is where Java is installed in your z/OS environment
    - MQPATH is where the MQ binaries are accessible from
    - WLPUSER is where the server directory will be created

```
//CRTMQWEB JOB 'MQ WEB',CLASS=A,REGION=0M,MSGCLASS=H,
// NOTIFY=&SYSUID
//*************************************************************
//*  SET SYMBOLS
//*************************************************************
//EXPORT EXPORT SYMLIST=(*)
// SET JAVAHOME='/usr/lpp/java/J8.0_64'
// SET MQPATH='/usr/lpp/mqm/V9R4MX/web'
// SET WLPUSER='/var/mqm'
//*************************************************************
//*  Step crtmqweb - Use the crtmqweb command
//*************************************************************
//CRTMQWEB EXEC PGM=IKJEFT01,REGION=0M
//SYSTSPRT DD SYSOUT=*
```

```
//SYSERR   DD SYSOUT=*
//STDOUT   DD SYSOUT=*
//SYSTSIN  DD *,SYMBOLS=EXECSYS
 BPXBATCH SH +
 export JAVA_HOME=&JAVAHOME; +
 export WLP_USER_DIR=&WLPUSER; +
 &MQPATH/bin/crtmqweb &WLPUSER -p MQ
```

5. Submit the JCL by typing “submit” in the command line and pressing **Enter**.



6. Next, let’s check out the angel process. What is that? The Liberty angel process is a started task that allows Liberty servers to use z/OS authorized services. It's long-lived and can be shared among your multiple Liberty servers. Use the F3 key to back out of CRTMQWEB and place an ‘e’ to the left of the MQANGEL member, then hit enter.

Sample JCL for MQANGEL:

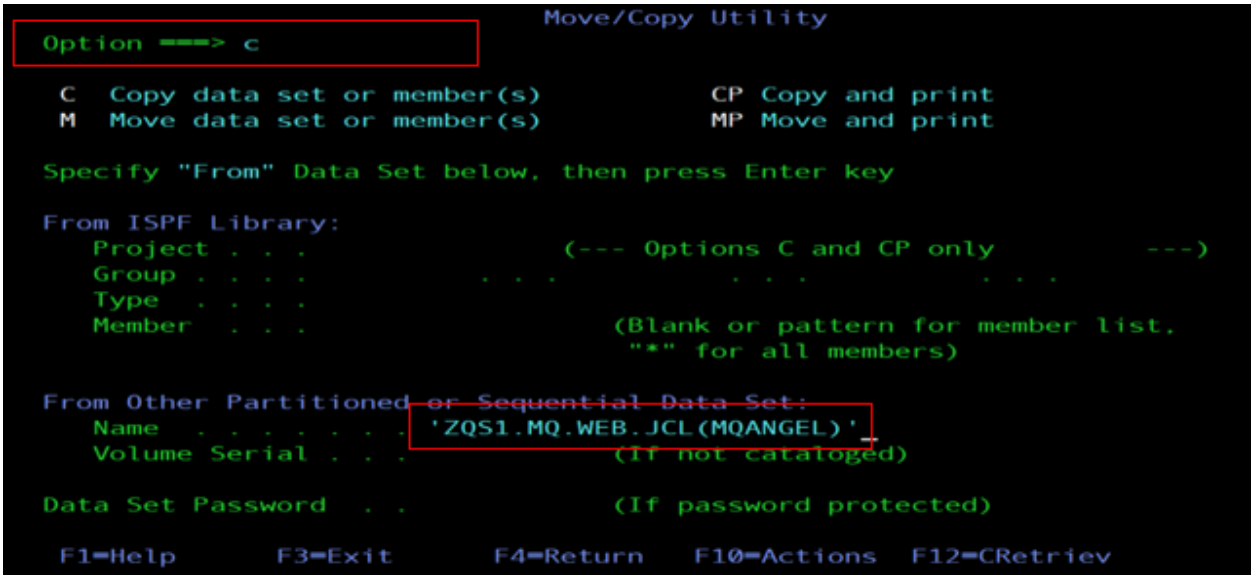
```
//MQANGEL PROC PARMS='',COLD=N,NAME='MQANGEL',SAFLOG=Y
//*-------------------------------------------------------------
//  SET ROOT='/usr/lpp/mqm/V9R4MX/web'
//*-------------------------------------------------------------
//* Start the Liberty angel process for MQ
```

```
//*-------------------------------------------------------------------
//STEP1   EXEC PGM=BPXBATA2,REGION=0M,TIME=NOLIMIT,
//     PARM='PGM &ROOT./lib/native/zos/s390x/bbgzangl COLD=&COLD NAME=X
//          &NAME &PARMS SAFLOG=&SAFLOG'
```

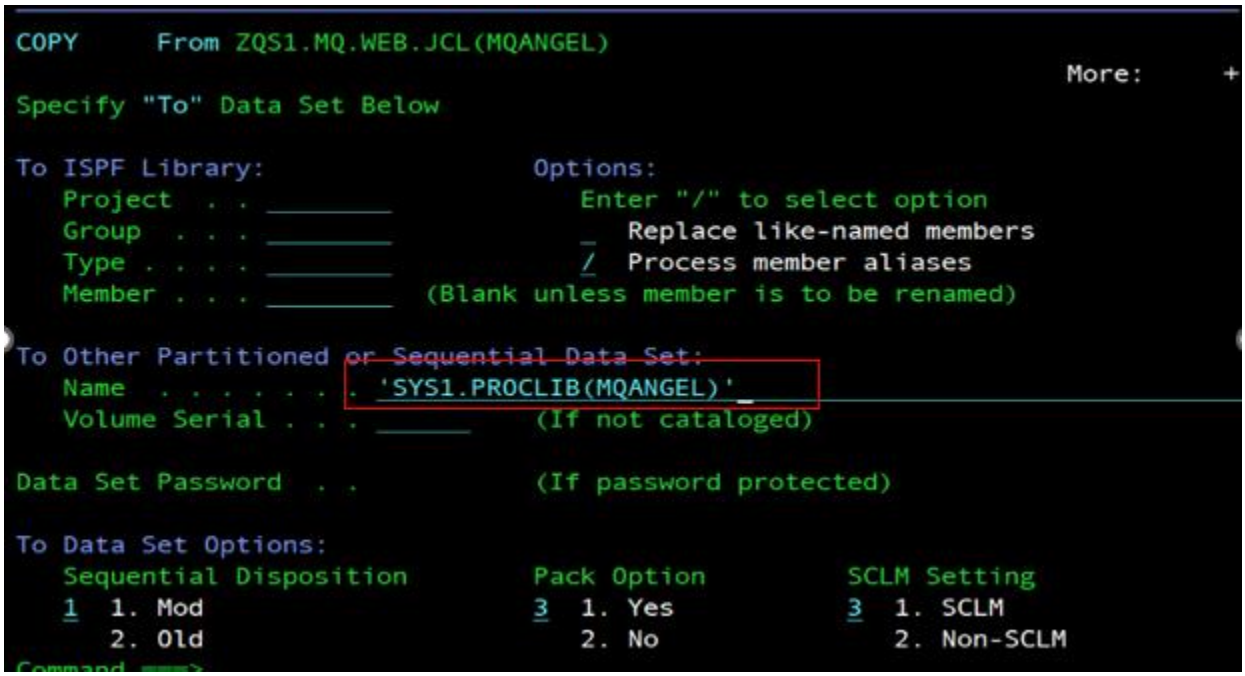
7. Lastly, let's check out the MQWEBS JCL. Using the 'e' edit function, make sure the JCL matches the below :

```
****** ******************************** Top of Data ********************************
==MSG> -Warning- The UNDO command is not available until you change
==MSG>           your edit profile using the command RECOVERY ON.
000001 //MQWEBS PROC PARMS='mqweb --clean'
000002 //*
000003 //   SET INSTDIR='/usr/lpp/mqm/V9R4MX/web'
000004 //   SET USERDIR='/var/mqm'
000005 //*
000006 //MQCONSOL EXEC PGM=BPXBATSL,REGION=0M,TIME=NOLIMIT,
000007 //   PARM='PGM &INSTDIR./lib/native/zos/s390x/bbgzsrv &PARMS.'
000008 //WLPUDIR  DD PATH='&USERDIR.'
000009 //STEPLIB  DD DSN=MQ940CD.SCSQANLE,DISP=SHR
000010 //         DD DSN=MQ940CD.SCSQAUTH,DISP=SHR
000011 //STDOUT   DD SYSOUT=*
000012 //STDERR   DD SYSOUT=*
000013 //STDIN    DD DUMMY
000014 //STDENV   DD *
000015 JAVA_HOME=/usr/lpp/java/J8.0_64
000016 WLP_USER_DIR=/var/mqm/
000017 PATH=/usr/lpp/mqm/V9R4MX/web/bin:/bin:/usr/bin
000018 LIBPATH=/usr/lpp/mqm/V9R4MX/java/lib
000019 //ZTDENV   DD *
000020 JAVA_HOME=/usr/lpp/java/J8.0_64
000021 WLP_USER_DIR=/var/mqm/
000022 PATH=/usr/lpp/mqm/V9R4MX/web/bin:/bin:/usr/bin
000023 LIBPATH=/usr/lpp/mqm/V9R4MX/java/lib
000024 IBM_JAVA_OPTIONS=-Dcom.ibm.ws.zos.core.angelName=MQANGEL
****** ***************************** Bottom of Data ********************************
```

8. We won't be submitting MQANGEL or MQWEBS, instead we will be adding it to the SYS1.PROCLIB. SYS1.PROCLIB is a system library in which the procedures that are included with the product are placed when you run the starter job.
9. Back out of MQANGEL and navigate to option 3.3 from the main menu. Enter the option 'C' for copy and then specify the member you'd like to copy next to 'Name'. NOTE! You must use single quotes around the data set names.



10. You want the destination for the copy to be 'SYS1.PROCLIB(MQANGEL)' . Specify this here.



11. Repeat this copying process for 'ZQS1.MQ.WEB.JCL(MQWEBS)'
12. Excellent! Now both MQANGEL and MQWEBS will be included in the SYS1.PROCLIB as started tasks. You can navigate to SYS1.PROCLIB using 3.4 if you'd like to confirm!

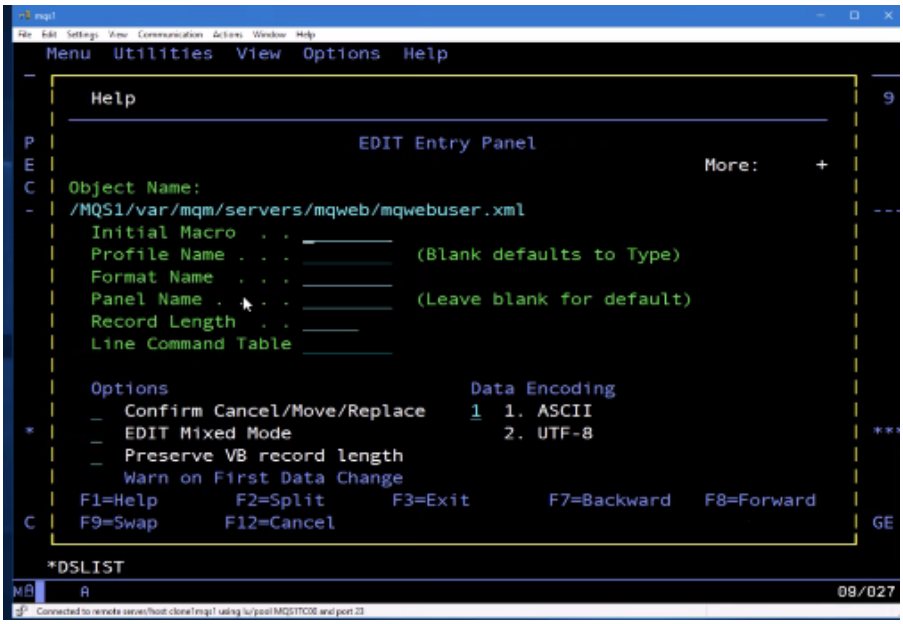
Step 2. Modify the XML files

In order to configure our web console, we will customize the Websphere Liberty server spun up by running CRTMQWEB. The XML that is used to construct the web console is located in the /var/mqm directory.

1. Navigate to the /var/mqm/ directory from option 3.4 in ISPF. In the servers directory, several XML files were created. You need to modify these files.
2. Put an 'l' to the left of the 'servers' option to browse its contents. Repeat for the 'mqweb' directory until you see several XML files.

```
<?xml version="1.0" encoding="UTF-8"?>
 <server>
   <featureManager>
     <feature>appSecurity-2.0</feature>
   </featureManager>
   <webAppSecurity allowFailOverToBasicAuth="true"/>
   <variable name="httpsPort" value="9443"/>
   <variable name="httpHost" value="-1"/>
   <variable name="mqRestMessagingEnabled" value="true"/>
   <httpEndpoint host="*" httpPort="-1" httpsPost="9443"
     id="defaultHttpEndpoint"/>
</server>
```

3. Type "ea" next to mqwebuser.xml to open the XML file in edit mode. An 'EDIT Entry Panel', like the one shown below will pop up, press **Enter** to move past it.



4. Once in edit mode, copy and paste the following lines into the file. Save the XML file and back out.

```
<?xml version="1.0" encoding="UTF-8"?>
 <server>
   <featureManager>
     <feature>appSecurity-2.0</feature>
   </featureManager>
   <webAppSecurity allowFailOverToBasicAuth="true"/>
   <variable name="httpsPort" value="9443"/>
   <variable name="httpHost" value="-1"/>
   <variable name="mqRestMessagingEnabled" value="true"/>
   <httpEndpoint host="*" httpPort="-1" httpsPost="9443"
     id="defaultHttpEndpoint"/>
</server>
```

5. Edit the sever.xml file. Add this line: `<include location="basic_registry.xml"/>`. Save the XML file and back out.

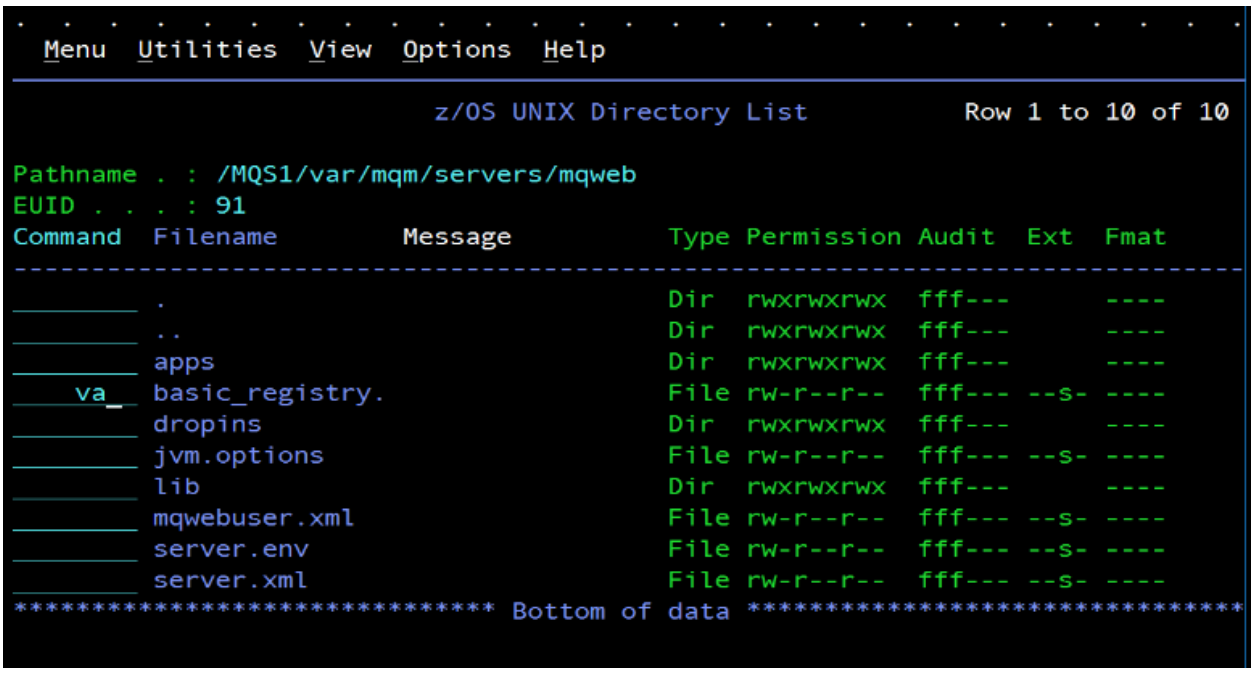
6. To create this basic_registry.xml file, pull the sample XML file from the MQ installation.
   a. Navigate to /usr/lpp/mqm//V9RXMX/web/mq/samp/configuration using 3.4 from the ISPF menu.
   b. Type 'tso omvs' in your command line.
   c. In OMVS, type 'ls /usr/lpp/mqm/V9RXMX/web/mq/samp/configuration'. When you press **Enter**, you should see a list of the XML files in the directory, including basic_registry.xml.

```
DQUINCY:/Z31RA1/usr/lpp/mqm/V9R4MX/web/mq/samp/configuration: >ls
basic_registry.xml     no_security.xml
ldap_registry.xml      zos_saf_registry.xml
```

Change directories to `/var/mqm/servers/mqweb` and then execute this command:

```
cp /usr/lpp/mqm/V9R4MX/web/mq/samp/configuration/basic_registry.xml .
```

7. Enter ‘exit’ to quit out of OMVS. Back out of the /MQS1/var/mqm/servers/mqweb directory, and then re-enter it. Browse the basic_registry.xml file using ‘va’ to look at what credentials users will be able to use for the web console. Make note of the mqadmin username and password. You will need those later.



```
/MQS1/var/mqm/servers/mqweb/basic_registry.xml  Columns 00001 00072
Sample Basic Registry
-->
<basicRegistry id="basic" realm="defaultRealm">
    <!--
    This sample defines two users with unencoded passwords
    and a group, these are used by the role mappings above.
    -->
    <user name="mqadmin" password="mqadmin"/>
    <user name="mqreader" password="mqreader"/>
    <user name="mftadmin" password="mftadmin"/>
    <user name="mftreader" password="mftreader"/>
    <group name="MQWebAdminGroup">
       <member name="mqadmin"/>
    </group>
</basicRegistry>

<!--
Enable HTTPS on a specific port by uncommenting the line below and p
-->
```

Step 3. Access the MQ console

Now that you are all configured, start up the console!

1. From the ISPF main menu, type 'sdsf' and press **Enter**. On the new popup menu, type a '/' in the command line and press **Enter**.

```
 Display  Filter  View  Print  Options  Search  Help
------------------------------------------------------------------------------
SDSF MENU 3.1    MQPLEX1   MQS1                        LINE 1-19 (94)
NP    NAME     Description              Group    Status
      DA       Active users             Jobs
      I        Input queue              Jobs
      O        Output queue             Output
      H        Held output queue        Output
      ST       Status of jobs           Jobs
      JG       Job groups               JES
      SYM      System symbols           System
      LOG      System log               Log
      SR       System requests          Log
      MAS      Members in the MAS       JES
      JC       Job classes              JES
      SE       Scheduling environments  WLM
      RES      WLM resources            WLM
      ENC      Enclaves                 WLM
      PS       Processes                OMVS
      SYS      System information       System
      ENQ      Enqueues                 System
      ENQC     Enqueue contention       System
COMMAND INPUT ===> /_                                 SCROLL ===> CSR
```

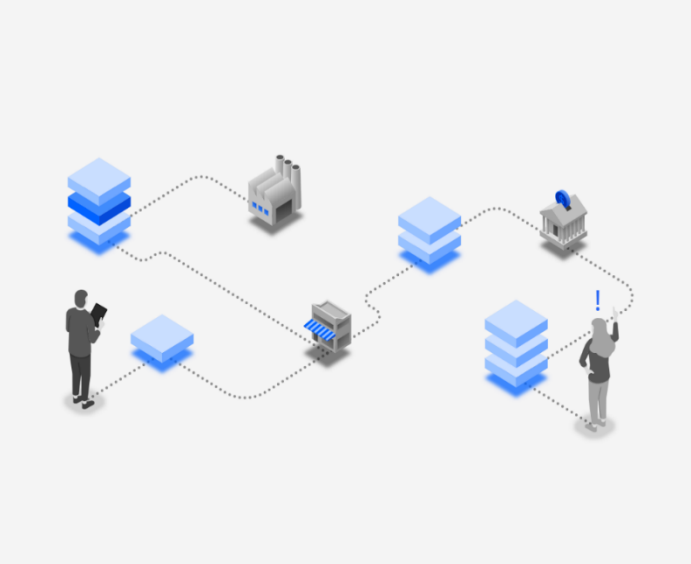
2. In the command window, type the command ‘s mqwebs’ to start the MQ console.
    a. You can take down the console by using the command ‘p mqwebs’ from the SDSF command shell

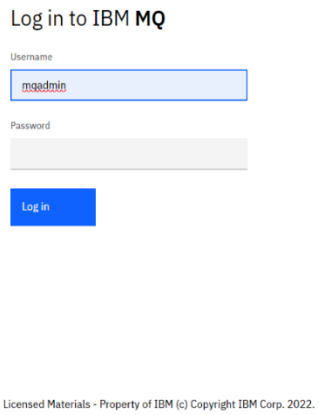
If you navigate to ‘DA’ from the SDSF main menu, you will see ‘MQWEBS’ is now running!

Open a browser (Chrome is recommended) in your virtual machine, and go to the following web address: https://zos_ip_address:9443/ibmmq/console where X is your lab assigned z/OS environment.

    b. Tech tip: Troubleshoot any errors in the MQ web console itself by using the logs provided by the liberty server.
3. On the MQ Console login screen, use the credentials you saw in the basic_registry.xml to login (The default credentials are mqadmin/mqadmin)

a. Tech tip: Troubleshoot using the ‘ST’ function of SDSF, you may have a JCL error with MQWEBS if the console doesn’t start up with the ‘s mqwebs’ command



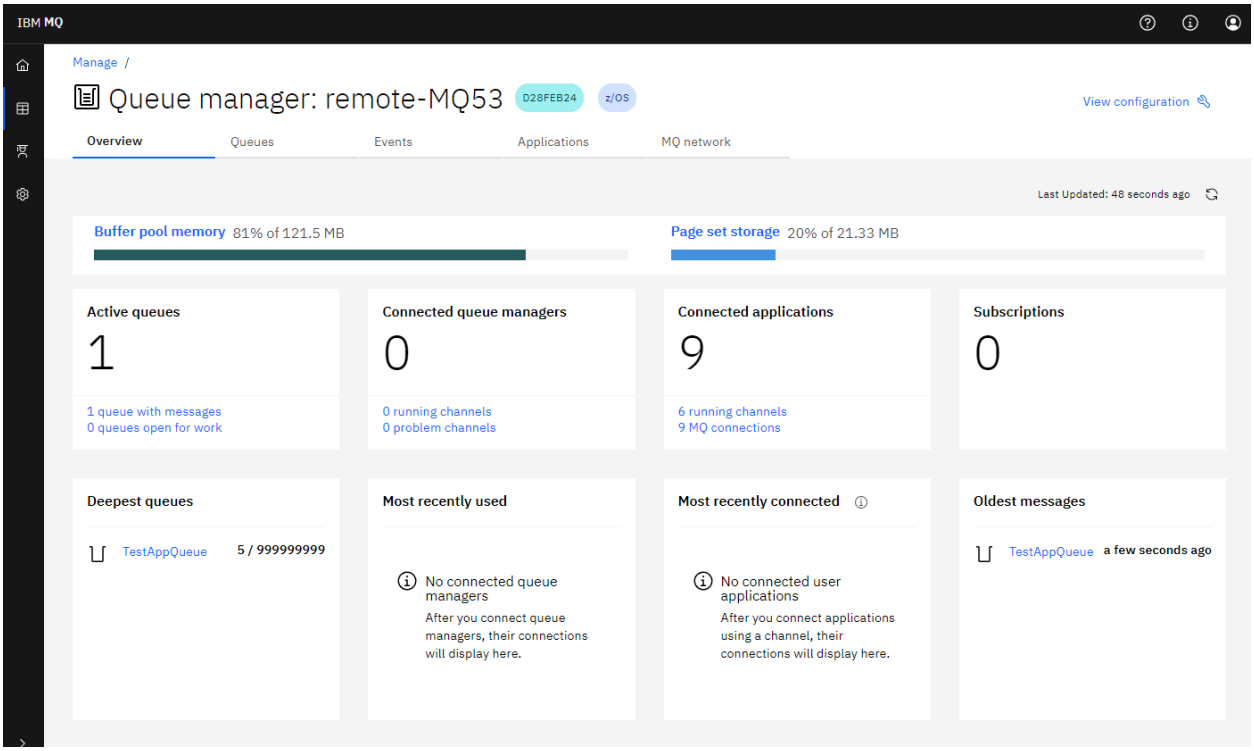


Step 4. View and edit MQ objects by using the MQ Console

4. If no queue managers are running, go back to your z/OS environment and from the ‘sdsf’ command shell, enter the commands, where +cpf is your queue manager name

```
+cpf START QMGR
+cpf START CHINIT
```

5. From the Home page in the MQ console, select Manage to open a list of queue managers. You can create or manage them from here.
6. Selecting a queue manager will display the queues and other objects associated with this queue manager.
7. If you are using a queue manager with IBM MQ version 9.4 for z/OS, you will be able to have visibility into storage including ‘Storage classes’, ‘Buffer pools’ and ‘Page sets’.
8. Navigate to investigate your buffer pools and page set utilization to determine which queues are consuming the most memory and storage



Summary and next steps

Nice, you now have an MQ web console set up on your z/OS environment! If you are not able to access a test z/OS environment, please reach out to me at dorothy.quincy@ibm.com to provision a sample lab environment in the Washington Systems Center.

The next step here is to tighten the security by upgrading from basic authentication to using the SAF registry. You can find more information about that here.

I'd like to thank Mitch Johnson in the Washington Systems Center for developing the JCL used in this tutorial.