WebSphere® Message Broker V6

Configuration and Customization

Note: In the following document, the name CSQ6 is being used as the WebSphere MQ queue manager name and the prefix for broker, Configuration Manager. In any examples, your chosen naming conventions would replace these names.

1. Environment Set-up

The prerequisite products for WebSphere Message Broker V6 on z/OS® are:

- ✓ z/OS V1.5 or later (with APAR OA11699)
- ✓ WebSphere Message Broker V6
- ✓ WebSphere MQ V5.3.1 (+ PTFS) or later
- ✓ DB2® V7.1 (+PTFs) or later
- ✓ Java™ V1.4.2
- ✓ XML Toolkit v1.8

NOTE: Always check the WebSphere Message Broker V6 website for latest information on prerequisite product levels at http://www-306.ibm.com/software/integration/wbimessagebroker/v6/zos.html

In order to customize a broker component complete the following steps.

- ✓ Install WebSphere Message Broker V6 for z/OS; mount the broker HFS, example: ,/usr/lpp/mqsi
- ✓ Install WebSphere MQ for z/OS; mount the WebSphere MQ HFS, for example, /usr/lpp/mqm
- ✓ Install XML Toolkit V1.8; mount the HFS, for example, /usr/lpp/ixm
- ✓ Install Java V1.4.2; mount HFS, for example, /usr/lpp/java
- ✓ Set bipimain attributes to authorized
- ✓ Set USS BPXPRMxx MAXxxxxx PARMLIB attributes per documentation and your environment
- ✓ Define RACF started task userids and group ids, for example:
  - group MQSI
  - userid CSQ6BRK, with OMVS (/u/CSQ6BRK), no TSO, group MQSI
  - same for CSQ6CFG
- ✓ Define component directories, for example
  - /var/mqsi/CSQ6BRK and /var/mqsi/CSQ6CFG
  - check authorities for group (for example, MQSI)
    - id CSQ6BRK (to obtain the group set of the user ID)
    - chgrp –R MQSI /var/mqsi/CSQ6BRK
    - chmod g=rwx /var/mqsi/CSQ6BRK
  - set up RACF STARTED class for each component with alter authority
- ✓ Set up WebSphere MQ queue manager for use by component (QMgr must be unique for each broker; Configuration Manager may share a QMgr with the broker)
- ✓ Give component started task userid(s) (for example, CSQ6BRK, CSQ6CFG), appropriate authorities:
  - rx to runtime libraries (for example,/usr/lpp/mqsi)
  - rwx to component directories (for example, /var/mqsi/CSQ6BRK, /var/mqsi/CSQ6CFG)
  - rw to home directory (for example, /u/CSQ6BRK, /u/CSQ6CFG)
  - DB2 SELECT on tables SYSIBM.SYSTABLES, SYSIBM.SYSSYNONYMS, and SYSIBM.SYSDATABASE
  - DB2 SELECT, UPDATE, INSERT, and DELETE privileges on all broker system tables

  - DB2_TABLE_OWNER a valid authorization ID of the started task user ID
  - DB2 EXECUTE authority on DSNACLI plan

✓ Ensure that the broker, Java runtime, ODBC, and z/OS Language Environment are XPLink-compliant

2. Determine Customization Information

Before you begin to set up a broker, user name server, or Configuration Manager, you must determine the customization information that you will use in your environment. As in V5 of the broker, this information must be discussed with your system, DB2 and WebSphere MQ administrators (if different than yourself). This information is what is used to build your broker and Configuration Manager PDSEs.

The names in the example below should be changed to match the HLQs, file system directories, PDSE names in your enterprise.

NOTE: this is the way the planning information is laid out in the documentation. The fields in the BIPEDIT file that you will be updating are not necessarily in the same order as the items in this planning information and there is not necessarily a one-to-one comparison. However, in customizing your JCL, you should find all of the needed ++…++ fields from your planning template.

Installation information for broker

HLQ of SBIPPROC		MQSERIES.WBI.V6R0M0
HLQ of SBIPSAMP		MQSERIES.WBI.V6R0M0
HFS install directory	++INSTALL++	/usr/lpp/mqsi/V6R0M0
LE HLQ	++LEHQL++	CEE
WebSphere MQ HLQ	++WMQHLQ++	MQSERIES.V6R0M0
System HLQ	++SYSHLQ++	SYS1
Java install directory	++JAVA++	/usr/lpp/java/J1.4
Local environment	++LOCALE++	C
Time zone	++TIMEZONE++	CST6

Installation information for Configuration Manager
Same as above plus

WebSphere MQ JMS install directory	++MQPATH++	/usr/lpp/mqm/V6R0M0

DB2 information for broker

HLQ	++DB2HLQ++	DB2.V8R1M0
run library	++DB2RUNLIB++	DB2WK8.RUNLIB.LOAD
Subsystem id	++DB2SUBSYSTEM++	DSN8
Plan name	++DB2DSNACLIPLAN++	DSNACLI
SampleProgram Name	++DB2SAMPLEPROGRAM++	DSNTEP2
Sample Program Plan	++DB2SAMPLEPROGRAMPLAN++	DSNTEP81
Location	++DB2LOCATION++	**see note below**
Conversion	++DB2CONVERSION++	SINGLE
Table owner user ID	++DB2TABLEOWNER++	CSQ6BRK

NOTE: The DB2 location is available by entering command –DSN8 (replace using your DB2 command prefix) DIS DDF from the console (or ISPF SDSF).

Component information for broker

User ID directory	++HOME++	/u/CSQ6BRK
QMgr	++QUEUEMANAGER++	CSQ6
HFS directory	++COMPONENT DIRECTORY++	/var/mqsi/CSQ6BRK
Broker name	++COMPONENTNAME++	CSQ6BRK
JCL dataset	++COMPONENTDATASET++	WMBV6.CSQ6BRK.CNTL
Profile	++COMPONENTPROFILE++	BIPBPROF
Started task	++STARTEDTASKNAME++	CSQ6BRK
Broker db2	++DB2DATABASE++	CSQ6BRK
DB2 storage	++DB2STORAGEGROUP++	CSQ6BRK
DB2 buffer	++DB2BUFFERPOOL++	BP0
mqsicreatebroker options	++OPTIONS++	‘’ (that’s two consecutive single quotes)

Component information for Configuration Manager

Name	++COMPONENTNAME++	CMGR
User ID directory	++HOME++	/u/CSQ6CFG
QMgr	++QUEUEMANAGER++	CSQ6
HFS directory	++COMPONENTDIRECTORY++	/var/mqsi/CSQ6CFG
JCL dataset	++COMPONENTDATASET++	WMBV6.CSQ6CFG.CNTL
Profile	++COMPONENTPROFILE++	BIPCPROF
Started task	++STARTEDTASKNAME++	CSQ6CFG
mqsicreatconfigmgr options	++OPTIONS++	‘’ (two consecutive single quotes)

3. Create the broker

Again, the names in the example below should be changed to match the HLQs, file system directories, PDSE names in your enterprise.

3.1 Create PDSE

Allocate WMBV6.CSQ6BRK.CNTL
PO, FB, 80, 27920, Library, TRKS(15,1)

3.2 Customize the broker JCL

Below are the members you need to create a broker. However, it may be easier for you to simply copy all of the members from MQSERIES.WBI.V6R0M0.SBIPSAMP and SBIPPROC to your dataset. For a list of all members associated with the broker and an explanation of their use, see:

http://publib.boulder.ibm.com/infocenter/wmbhelp/v6r0m0/index.jsp?topic=/com.ibm.etools.mft.doc/ae22020_.htm

- Copy into WMBV6.CSQ6BRK.CNTL the following members from MQSERIES.WBI.V6R0M0.SBIPSAMP:
  `BIPBPROF` (broker profile).
  `BIPDSNAO` (DB2 dsnaoini).
- Copy into WMBV6.CSQ6BRK.CNTL the following members from MQSERIES.WBI.V6R0M0.SBIPPROC:
  `BIPCRBK` - job to create a broker.
  `BIPCRDB` - job to create the DB2 storage group, database and table spaces.

`BIPEDIT` - JCL customization. Rename `BIPEDIT` to a unique name that identifies it to the current component; for example, `CSQ6EDBK`.
`BIPGEN` - generate `ENVFILE`.
`BIPBRKP` (started task). Rename **`BIPBRKP`** to **`your broker started task name.`**

- After you have copied the members to you PDSE, customize the BIPEDIT member using the installation information collected above. Note if no options are needed, change ++OPTIONS++ line to two consecutive single quotes. Options are not needed if the ID submitting the BIPCBRK command has appropriate DB2 and WMQ authorities.

- A sample BIPEDIT member follows (in this example, the QMgr is CSQ6 and the broker prefix is the same):

```
/* REXX  */
/************************************************************************/
/*   See the product documentation on the meaning of the fields below=   */
/************************************************************************/
ISREDIT MACRO NOPROCESS
ADDRESS ISREDIT
/* All components (Broker , Configuration Manager and User Name Server) */
"change ++INSTALL++ /usr/lpp/mqsi/V6R0M0                all"
"change ++COMPONENTDIRECTORY++ /var/mqsi/CSQ6BRK         all"
"change ++COMPONENTNAME++ CSQ6BRK                    all"
"change ++HOME++ /u/CSQ6BRK                          all"
"change ++OPTIONS++ ''                             all"
"change ++LOCALE++ C                               all"
"change ++TIMEZONE++ GMT6CST                         all"
"change ++JAVA++ /usr/lpp/java/J1.4                   all"
"change ++LEHLQ++ CEE                              all"
"change ++WMQHLQ++ MQSERIES.V6R0M0                   all"
"change ++QUEUEMANAGER++ CSQ6                      all"
"change ++COMPONENTDATASET++ WMBV6.CSQ6BRK.CNTL          all"
"change ++STARTEDTASKNAME++ CSQ6BRK                  all"
"change ++COMPONENTPROFILE++ BIPBPROF                 all"
"change ++XMLTOOLKIT++ /usr/lpp/ixm/IBM/xml4c-5_5        all"
/* Broker only                                     */
/* (Delete if Configuration Manager or User Name Server)              */
"change ++DB2CONVERSION++ SINGLE                    all"
"change ++DB2SUBSYSTEM++ DSN8                      all"
"change ++DB2LOCATION++ MVS227D1                    all"
"change ++DB2TABLEOWNER++ CSQ6BRK                   all"
"change ++DB2CURRENTSQLID++ CSQ6BRK                  all"
"change ++DB2DSNACLIPLAN++ DSNACLI                   all"
"change ++DB2HLQ++ DB2.V8R1M0                      all"
"change ++DB2RUNLIB++ DB2WK8.RUNLIB.LOAD               all"
"change ++DB2SAMPLEPROGRAM++ DSNTEP2                 all"
"change ++DB2SAMPLEPROGRAMPLAN++ DSNTEP81              all"
"change ++DB2DATABASE++ CSQ6BRK                     all"
"change ++DB2STORAGEGROUP++ CSQ6BRK                  all"
"change ++DB2BUFFERPOOL++ BP0                      all"
"change ++DB2INDEXBP++ BP0                        all"
"change ++DB2LOBBP++ BP0                         all"
/* Configuration Manager only                          */
/* (Delete if Broker or User Name Server)                   */
"change ++MQPATH++ /usr/lpp/mqm/V6R0M0                 all"
□
```

- Rename your BIPEDIT member to something unique (for example, CSQ6BKED).

- Activate your renamed BIPEDIT file by running the following TSO command:

  ```
  ALTLIB ACTIVATE APPLICATION(EXEC) DA('WMBV6.CSQ6BRK.CNTL')
  ```

- Use ISPF to edit the dataset; edit each member and run your renamed BIPEDIT exec by typing its name on the command line.

- Put installation acceptable job card on each job. Note that your renamed BIPEDIT file is a REXX exec so no job card is needed. Likewise, your renamed BIPBRKP does not need a job card since it is a started task. BIPBPROF and BIPDSNAO do not take job cards.

3.3 Create the environment file

- Review BIPBPROF member for correctness
- Run member BIPGEN to create the environment file

NOTE: BIPGEN must be run any time you changed the BIPBPROF.

3.4 Prime DB2

- Ensure that DB2 location is correct; display using
  - –DSN8 DIS DDF (where –DSN8 is the DB2 command prefix)
  - this is what is used in DSNACL0 member
- Grant broker userid 'execute' on DSNACLI plan
- Check DSNACLI for asterisks

- Modify BIPCRDB to adjust LOB settings
  - edit the BIPCRDB job to modify the size of table space allocations to accommodate your broker

  - BIPCRDB will be changed so that the VCAT is equal to the specified ++LOCATION++; if not what you want the HLQ of the database datasets to be, modify BIPCRDB to change VCAT to HLQ wanted and volume wanted

- Run BIPCRDB; ensure cc zeros and table spaces created

- Grant broker userid 'insert, select, update' on all broker tables

Following are sample input commands to DB2 to accomplish the needed DB2 authorities for the broker ID; in this example, the broker ID is CSQ6BRK and the DB2 table HLQ is CSQ6BRK as specified in the VCAT statement of the create database:

```
GRANT EXECUTE ON PLAN DSNACLI TO CSQ6BRK;
GRANT SELECT ON TABLE SYSIBM.SYSSYNONYMS TO CSQ6BRK;
GRANT SELECT ON TABLE SYSIBM.SYSTABLES TO CSQ6BRK;
GRANT SELECT ON TABLE SYSIBM.SYSDATABASE TO CSQ6BRK;
GRANT DELETE, INSERT, SELECT, UPDATE
   ON TABLE
   CSQ6BRK.BACLELOB,
   CSQ6BRK.BACLENTR,
   CSQ6BRK.BAGGREGA,
```

```
  CSQ6BRK.BAGGRLOB,
  CSQ6BRK.BCLIELOB,
  CSQ6BRK.BCLIENTU,
  CSQ6BRK.BGROUPNA,
  CSQ6BRK.BMQPSLOB,
  CSQ6BRK.BMQPSTOP,
  CSQ6BRK.BMULTICA,
  CSQ6BRK.BMULTLOB,
  CSQ6BRK.BNBRCONN,
  CSQ6BRK.BPUBLISH,
  CSQ6BRK.BPUBLLOB,
  CSQ6BRK.BRETAINE,
  CSQ6BRK.BRETLOB1,
  CSQ6BRK.BRETLOB2,
  CSQ6BRK.BRMINFO,
  CSQ6BRK.BRMPHLOB,
  CSQ6BRK.BRMPHYSI,
  CSQ6BRK.BRMRTDDE,
  CSQ6BRK.BRMRTDIN,
  CSQ6BRK.BRMRTLOB,
  CSQ6BRK.BRMWFDIN,
  CSQ6BRK.BROKALOB,
  CSQ6BRK.BROKERAA,
  CSQ6BRK.BROKERAE,
  CSQ6BRK.BROKERRE,
  CSQ6BRK.BROKRLOB,
  CSQ6BRK.BSUBLOB1,
  CSQ6BRK.BSUBLOB2,
  CSQ6BRK.BSUBLOB3,
  CSQ6BRK.BSUBSCRI,
  CSQ6BRK.BTOPOLOG,
  CSQ6BRK.BUSERCON,
  CSQ6BRK.BUSERMEM,
  CSQ6BRK.BUSERNAM
      TO CSQ6BRK;
```

3.5 Create the broker component

- Run BIPCRBK
- Ensure that broker queues have been added
- Ensure cc zeros and DB2 tables updated
- Check STDOUT in the job log for command output (or check /var/mqsi/CSQ6BRK/output/stdout)

3.6 Copy started task to procedure library

Rename BIPBRKP member in your PDSE (for example,‘WMBV6.CSQ6BRK.CNTL(BIPBRKP)’) to your started task name (for example, CSQ6BRK) and copy to your production proclib.

3.7 Start broker

Before starting your broker, verify that the databases were allocated on the specified volume; check that the broker queues were created (in MQ panels).

From SDSF start the broker, for example:

```
/s CSQ6BRK
```

4. Create Configuration Manager

4.1 Customize the Configuration Manager JCL

Below are the members you need to create a Configuration Manager. However, it may be easier for you to simply copy all of the members from MQSERIES.WBI.V6R0M0.SBIPSAMP and SBIPPROC to your dataset. For a list of all members associated with the broker and an explanation of their use, see:

http://publib.boulder.ibm.com/infocenter/wmbhelp/v6r0m0/index.jsp?topic=/com.ibm.etools.mft.doc/ae22040_.htm

- Copy the following JCL from MQSERIES.WBI.V6R0M0.SBIPSAMP to WMBV6.CSQ6CFG.CNTL:
  `BIPCPROF` (Configuration Manager profile).

- Copy the following JCL from MQSERIES.WBI.V6R0M0.SBIPPROC to WMBV6.CSQ6CFG.CNTL
  `BIPCRCM` - job to create a Configuration Manager.
  `BIPEDIT` - edit macro for customization. Rename `BIPEDIT` to a unique name that identifies it to the current component; for example, `CSQ6EDCM`.
  `BIPGEN` - job to generate the `ENVFILE`.
  `BIPCMGRP` – started task (rename to your Configuration Manager started task)

- Customize the BIPEDIT member using the installation information collected; note if no options specified, change to ‘’ (options not needed if the ID submitting the BIPCRCM command has appropriate WMQ authorities) .

- Rename your BIPEDIT member to something unique (for example, CSQ6EDCM)

- Activate your renamed BIPEDIT file by running the following TSO command:

  ```
  ALTLIB ACTIVATE APPLICATION(EXEC) DA(‘WMBV6.CSQ6CFG.CNTL’)
  ```

- Use ISPF to edit the dataset; edit each member and run your renamed BIPEDIT exec by typing its name on the command line.

- Put installation acceptable job card on each job. Note that your renamed BIPEDIT file is a REXX exec so no job card is needed. Likewise, your renamed BIPCMGRP does not need a job card since it is a started task. And BIPCPROF does not use a job card.

4.2 Create the environment file

- Review BIPCPROF member for correctness
- Run member BIPGEN to create the environment file

NOTE: BIPGEN must be run any time you changed the BIPCPROF.

4.3 Create the Configuration Manager component

- Run BIPCRCM
- Ensure that broker queues have been added
- Ensure cc zeros
- Check STDOUT for command output

4.4 Copy started task to procedure library

Rename BIPCMGRP member in your PDSE (for example,‘WMBV6.CSQ6BRK.CNTL(BIPCMGRP)’) to your started task name (for example, CSQ6CFG) and copy to your production proclib.

4.5 Start Configuration Manager

From SDSF start the Configuration Manager:

```
/s CSQ6CFG
```

5. Toolkit to Configuration Manager and deployment

5.1 General comments

The toolkit still needs to be used to connect to the Configuration Manager and create the brokers in the domain. To connect to the z/OS queue manager you need to have the WebSphere MQ chinit and listener running. You also need to have the WebSphere MQ Client Attach facility installed, as without this it will not be possible to connect to the Configuration Manager's queue manager directly. If the C.A.F. is not installed then a connection between the toolkit and the z/OS Configuration Manager can be made using an intermediate queue manager on Microsoft® Windows® and the correct channels defined between it and the z/OS queue manager.

Note: If WebSphere MQ security is enabled for the Configuration Manager queue manager, the correct authority will need to be setup to allow the remote Windows user to connect using the serverconn channel.

.Once a broker is defined in the Configuration Manager, BAR files can be deployed directly without the need for using the toolkit by using mqsideploy. Execution groups can be created and deleted also by using commands.

mqsideploy points to a file in the filesystem (HFS or zFS). This file is a BAR file which has been created in the toolkit, exported, and FTP'd as binary to z/OS.

5.2 Set up connectivity between Toolkit and Configuration Manager

To set up a broker domain for your z/OS Configuration Manager on your toolkit:

- Verify that your QMgr and CHINIT are running and the CHINIT is listening on the appropriate port. Verify that your Configuration Manager is running.

- If not using WebSphere MQ Client Attach feature which connects through the server connection channel SYSTEM.BRK.CONFIG, ensure that you have set up bidirectional WMQ channels and a transmission queue connecting the Windows queue manager to the z/OS queue manager.

- Ensure that your toolkit machine/id has been given the appropriate authorization on the z/OS Configuration Manager.

  In SDSF, to grant access to your user ID for all machines enter, for example:
  '/F CSQ6CFG CA U=CSQ6CFG,A=YES,P=YES,X=F'
  or to grant access for a specific machine enter, for example:
  '/F CSQ6CFG CA U=CSQ6CFG,M=TABLE22x,P=YES,X=F'

  Note: A=YES means all machines. P=YES means all resources (a bit like superuser access)

- Verify by entering: /F CSQ6CFG,LA

- On toolkit, go to Broker Administration Perspective and in domain panel, create a domain connection (right click, new➔ domain)

- Click finish; the Configuration Manager should then appear in your domain window; check syslog messages on z/OS.

You add a broker to the z/OS Configuration Manager using the toolkit in the same manner as you would to any other Configuration Manager. In the Domain window, right click, select new broker and when presented with the window, select your z/OS Configuration Manager and put in the z/OS broker name and QMgr name.

5.3 Send BAR files to z/OS Configuration Manager and deploy

To send a BAR from the toolkit to a z/OS Configuration Manager, export the BAR to a file. FTP the BAR file, in binary, to the z/OS system. Put the BAR file into a filesystem (HFS or zFS).

From toolkit, export a bar file. (for example, to \bardir\bar1)
From Windows command line:

```
cd \bardir
ftp your-ipaddress
login userid with password
bin
put bar1 /u/CSQ6CFG/bar1
```

Deploy using the mqsideploy command. This is available on z/OS both as a console command (DP) and from JCL BIPDPLY.

For example, from SDSF enter: /f CSQ6CFG,dp b=CSQ6BRK a=/u/CSQ6CFG/bar1

This performs an incremental deployment. Add the m=yes parameter to perform a complete bar file deployment.