

Using CCDT in a Connection Factory for a Queue Manager Group of separate queue managers in WAS V7, V8.0 and V8.5

<https://www.ibm.com/support/pages/node/592499>

Date of last edition: 26-Sep-2013
Minor correction date: 10-Apr-2024

Angel Rivera - rivera@us.ibm.com
IBM WebSphere MQ Support

+++ Objective

This technical document describes the steps for configuring a Connection Factory in WebSphere Application Server V7 or V8.x that uses a Client Channel Definition Table (CCDT) file for a Queue Manager Group in WebSphere MQ V7 that use separate queue managers.

The last section has Information about customizable properties for Listener Ports.

If you want to use an Activation Specification, consult the following related techdoc:

<http://www.ibm.com/support/docview.wss?rs=171&uid=swg27035714>

Using CCDT in an Activation Specification for a Queue Manager Group of separate queue managers in WAS V7

This document explains the usage of the sample MDB (SampleMDBEJBEAR) which is available from the following techdoc, which also has the steps on how to create the JMS administrative objects and how to deploy the MDB.

Using WebSphere MQ V7 as JMS Provider for WebSphere Application Server V7

<http://www.ibm.com/support/docview.wss?rs=171&uid=swg27016505>

For the case of a Queue Manager Group that is composed of a multi-instance queue manager, see the techdoc:

<http://www.ibm.com/support/docview.wss?rs=171&uid=swg27020701>

Using a CCDT file to connect to WebSphere MQ multi-instance queue managers from WebSphere Application Server V7

Note: XA is NOT supported when you are trying to use DIFFERENT queue managers using CCDT, but if you are using the same queue manager such as in a multi-instance MQ queue manager, then it is supported.

+ tag: 7035452

+++++ Chapter 1: Configuration, creation of JMS administrative objects, deploy MDB +++++

++ There are 2 separate queue managers in the Queue Manager Group:

Queue Manager 1:
Name: QM_VER
Hostname: veracruz.x
Port: 1414
Channel: TEST.TO.QM_VER
Queue: Q_MDB

Queue Manager 2:
Name: QM_MIG
Hostname: veracruz.x
Port: 1421
Channel: TEST.TO.QM_MIG
Queue: Q_MDB

The WAS Server (server1) is located in:
Hostname: veracruz.x

The sample MDB code has been downloaded and it is available for deployment.

++ Creation of the CCDT

The CCDT is created in the queue manager QM_VER.

Notice that the queue manager group is the value for the attribute QMNAME:
QMGRP

Thus, to specify it, it is necessary to use a leading asterisk when using it:
*QMGRP

Invoke the MQ administration tool for the queue manager QM_VER

```
runmqsc QM_VER
```

```
DEFINE CHANNEL(TEST.TO.QM_VER) CHLTYPE(SVRCONN) TRPTYPE(TCP) REPLACE
```

```
DEFINE CHANNEL(TEST.TO.QM_MIG) CHLTYPE(CLNTCONN) TRPTYPE(TCP) CON-  
NAME('veracruz.x.(1421)') QMNAME(QMGRP) REPLACE
```

```
DEFINE CHANNEL(TEST.TO.QM_VER) CHLTYPE(CLNTCONN) TRPTYPE(TCP) CON-  
NAME('veracruz.x.(1414)') QMNAME(QMGRP) REPLACE
```

END

Invoke the MQ administration tool for the queue manager QM_MIG

In the other queue manager (QM_MIG) create the corresponding server-connection channel:

```
runmqsc QM_MIG
```

```
DEFINE CHANNEL(TEST.TO.QM_MIG) CHLTYPE(SVRCONN) TRPTYPE(TCP) REPLACE
```

END

+ Copy the CCDT file from QM_VER into a directory that is not associated with the queue manager in order to isolate disruptions from further updates to the client-connection channels in that queue manager. In this case it is /var/mqm
cp /var/mqm/QM_VER/@ipcc/AMQCLCHL.TAB /var/mqm/AMQCLCHL.TAB

The specification for WAS for the CCDT will be using this generic place.
/var/mqm/AMQCLCHL.TAB

++ Use the WAS Admin console to create JMS resources

This scenario is based on a single standalone WAS server (no other cells, no other nodes), thus, the scope used here is for the "server".

+ Connection Factory:

JMS messaging provider: WebSphere MQ Messaging Provider

Name: SampleMDBConnectionFactoryCCDT

JNDI Name: jms/SampleMDBConnectionFactoryCCDT

Use a client channel definition table:

CCDT: <file:///var/mqm/AMQCLCHL.TAB>

Queue Manager: *QMGRP

Notice that you will get the following warning:

The value *QMGRP for field Queue manager used in combination with a client channel definition table might prevent recovery of global transactions. See the information center for further detail.

IMPORTANT! XA is NOT Supported in SOME scenarios (but supported in others)

Note: As of 10-Apr-2024 the following article was removed from the web and it is no longer available:

<http://www-01.ibm.com/support/docview.wss?uid=swg21631879>
Configuring WebSphere Application Server to connect via a Client Channel Definition Table to a WMQ queue manager for Global Transaction (XA) capability.

This is the summary:

+ begin

If you are using a Client Channel Definition Table (CCDT) to connect from WebSphere Application Server to a WebSphere MQ queue manager, the following table shows the configurations in which XA is and is not supported:

These scenarios ARE NOT Supported:

- Using a CCDT to connect to multiple different queue managers (active/active)
- Using a CCDT to connect to multiple multi-instance queue managers (active/active)

These scenarios ARE Supported:

- CCDT connecting to single queue manager
- CCDT connecting to a single multi-instance queue manager (active/standby)

+ end

IMPORTANT! You must disable XA if you are trying to reconnect to a different queue manager

If you are trying to reconnect to a different queue manager, after you create the Connection Factory is ESSENTIAL that you perform the following change because XA is NOT supported for reconnection to different queue managers:

In the main page for the Connection Factory, scroll down to the "Advanced" section and UNCHECK the following attribute:

Support distributed two phase commit protocol

Click OK and then Save.

If you look at the corresponding entry for this connection factory in the resources.xml file, you will see that the attribute is called:

XAEnabled

A value of 'false' disables the use of XA.

+ Destination: Queue

JMS messaging provider: WebSphere MQ Messaging Provider

Name: SampleMDBQueueNoQmgr

JNDI Name: jms/SampleMDBQueueNoQmgr

Queue Name in MQ: Q_MDB

Queue Manager: (blank)

Note: It is ESSENTIAL that the field for the Queue Manager name must be left blank. If you look at the corresponding entry for this connection factory in the resources.xml file, you will see that the attribute is called:

baseQueueManagerName

+ Listener Port for Queue:

Name: SampleMDBQueueLPCCDT

Initial State: Started

Connection factory JNDI name: jms/SampleMDBConnectionFactoryCCDT

Destination JNDI name: jms/SampleMDBQueueNoQmgr

+ Restart the WAS server in order for the JMS resources to be in place

You will need to stop and restart the WAS server in order for the new JMS resources to be available

As root issue:

```
stopServer.sh server1  
startServer.sh server1
```

+ Deploy the MDB

From the WebSphere Application Server Administrative Console, in the left panel, select:

Applications > New Application

Then from the center window "New Application" select:
New Enterprise Application

You will see a dialog "Preparing for the application installation".
Specify the location where you have downloaded the ear file: SampleMDBEJB.ear

The sample MDB has a default binding that is associated with a specific Listener Port.
We need to override it now.

Do not select FastPath, instead select:

(*) Detailed - Show all installation options and parameters

In "Step 1: Select installation options", select:

(*) Deploy enterprise beans

Click on "Step 6: Bind listeners for message-driven beans"

In the right-most column (titled "Listener Bindings"), specify the following name:
SampleMDBQueueLPCCDT

Continue with the deployment of the MDB and wait for a successful deployment. Notice that the deployment will not automatically start the MDB, do not start it yet.

ADMA5013I: Application SampleMDBEJBEAR installed successfully.

Application SampleMDBEJBEAR installed successfully.

Click on Save.

+ Start the MDB

```
+++++
+++ Chapter 2: Testing of the MDB, including failover
+++++
```

+ Open a command window prompt to put a message into the queue for the queue manager QM_VER:

```
rivera@veracruz: /var/mqm
$ amqspout Q_MDB QM_VER
Sample AMQSPUT0 start
target queue is Q_MDB
test-msg-from-QM_VER
Sample AMQSPUT0 end
```

+ Change to the directory where the WAS SystemOut.log is located. In this test it is:
cd /opt/IBM/WebSphere/AppServer/profiles/AppSrv01/logs/server1

Issue:

```
tail -f SystemOut.log
```

Notice that the bottom line is provided by the MDB and displays the contents of the message:

```
... SystemOut    O +++ SAMPLE MDB: Text Message => test-msg-from-QM_VER
```

+ Perform the failover.

Now that we have successfully placed a message into QM_VER and the MDB has consumed it, let's stop QM_VER in order to trigger the reconnection to QM_MIG.

Issue:

```
endmqm -ir QM_VER
```

Notice the errors in the SystemOur.log

```
[7/10/12 16:06:10:467 EDT] 0000001e SibMessage W [:] CWSJY0003W: JMSSC3036:
An exception has been delivered to the connections exception listener: '
```

```
Message : com.ibm.msg.client.jms.DetailedJMSEException: JMS-
WMQ1107: A problem with this connection has occurred. An error has occurred with
the WebSphere MQ JMS connection. Use the linked exception to determine the cause
of this error.
```

```
Class : class com.ibm.msg.client.jms.DetailedJMSEException
```

```
...
```


Caused by [1] --> Message : com.ibm.mq.MQException: JMSCMQ0001: WebSphere MQ call failed with compcode '2' ('MQCC_FAILED') reason '2009' ('MQRC_CONNECTION_BROKEN').

Class : class com.ibm.mq.MQException

...

[7/10/12 16:06:10:481 EDT] 0000001f JMSEExceptionL E WMSG0018E: Error on JMSConnection for MDB SampleMDB , JMSDestination jms/SampleMDBQueueNoQmgr : com.ibm.msg.client.jms.DetailedJMSEException: JMSWMQ1107: A problem with this connection has occurred. An error has occurred with the WebSphere MQ JMS connection. Use the linked exception to determine the cause of this error.

at

com.ibm.msg.client.wmq.common.internal.Reason.reasonToException(Reason.java:608)

...

Caused by: com.ibm.mq.MQException: JMSCMQ0001: WebSphere MQ call failed with compcode '2' ('MQCC_FAILED') reason '2009' ('MQRC_CONNECTION_BROKEN').

...

[7/10/12 16:06:10:488 EDT] 0000001f JMSEExceptionL E WMSG0057E: Error on JMSConnection for MDB SampleMDB , JMSDestination jms/SampleMDBQueueNoQmgr , JMS Linked Exception : com.ibm.mq.MQException: JMSCMQ0001: WebSphere MQ call failed with compcode '2' ('MQCC_FAILED') reason '2009' ('MQRC_CONNECTION_BROKEN').

...

[7/10/12 16:06:10:650 EDT] 0000001f MCWrapper E J2CA0081E: Method cleanup failed while trying to execute method cleanup on ManagedConnection com.ibm.ejs.jms.JMSManagedConnection@6eef6eef

managed connection factory =

com.ibm.ejs.jms.WMQJMSRAManagedConnectionFactory@2e0c2e0c

physical connection = com.ibm.mq.jms.MQConnection@4bed4bed

connectorName =

JMS\$SampleMDBConnectionFactoryCCDT\$JMSManagedConnection@0

credential = null

open connection handles = [] from resource

jms/SampleMDBConnectionFactoryCCDT. Caught exception: java.

resource.spi.ResourceAdapterInternalException: Failed to stop connection on cleanup

...

Caused by: com.ibm.msg.client.jms.DetailedJMSEException: JMSCMQ0002: The method 'MQCTL' failed. A WebSphere MQ call failed. Please see the linked exception for more information.

...

Caused by: com.ibm.mq.MQException: JMSCMQ0001: WebSphere MQ call failed with compcode '2' ('MQCC_FAILED') reason '2009' ('MQRC_CONNECTION_BROKEN').

at

com.ibm.msg.client.wmq.common.internal.Reason.createException(Reason.java:223)

... 23 more

Caused by: com.ibm.mq.jmqi.JmqiException: CC=2;RC=2009

Notice that FFDC files are generated:

```
[7/10/12 16:06:10:520 EDT] 0000001f FfdcProvider W
com.ibm.ws.ffdc.impl.FfdcProvider logIncident FFDC1003I: FFDC Incident emitted on
/opt/IBM/WebSphere/AppServer/profiles/AppSrv01/logs/ffdc/server1_4d464d46_12.0
7.10_16.06.10.5074917.txt com.ibm.ejs.jms.listener.MDBListenerImpl.internalStop
358
```

Notice that attempt of the Listener Port to restart in 60 seconds:

```
[7/10/12 16:06:11:774 EDT] 0000001f MDBListenerIm I WMSG0043I: MDB Listener
SampleMDBQueueLPCCDT stopped for JMSDestination jms/SampleMDBQueueNoQmgr
[7/10/12 16:06:11:778 EDT] 0000001f MDBListenerIm I WMSG0058I: Listener Port
SampleMDBQueueLPCCDT will attempt to restart in 60 seconds
```

Notice that the Listener has restarted, it means that it was able to reconnect to the other queue manager (QM_MIG). In this case, the failover process took 62 seconds.

```
[7/10/12 16:07:12:617 EDT] 00000012 MDBListenerIm I WMSG0042I: MDB Listener
SampleMDBQueueLPCCDT started successfully for JMSDestination
jms/SampleMDBQueueNoQmgr
```

+ Now put a new message into QM_MIG (via "amqsput Q_MDB Q_MIG") and notice that the MDB is able to consume it:

```
[7/10/12 16:12:12:546 EDT] 0000001d SystemOut O +++ SAMPLE MDB: Text Message
=> test-msg-from-QM_MIG
```

YEAH! We were able to put messages in both queue managers and the MDB was able to read them, first from one queue manager, and then after the failover, from the other queue manager.

+++ Information about customizable properties for Listener Ports

You may want to customize the behavior of the Listener Port, such as how many times to retry to connect during a recovery (when a disconnect happens) and how long to wait between the retries.

http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/umb_prolscp.html

WebSphere Application Server (Distributed platforms and Windows), Version 7.0 >

Reference > Custom properties

File name: umb_prolscp.html

Message listener service custom properties

MAX.RECOVERY.RETRIES

The maximum number of times that a listener port managed by this service tries to recover from a failure before giving up and stopping. When stopped the associated listener port is changed to the stop state. The interval between retry attempts is defined by the RECOVERY.RETRY.INTERVAL custom property.

A failure can be one of two things:

- An unexpected error has occurred when a listener port tries to get a message from the JMS provider.
- The connection between the application server and the JMS provider has been lost, usually due to a network error.

Data type: Integer
Units: Retry attempts
Default: 5
Range: 0 (no retries) through 2147483647

RECOVERY.RETRY.INTERVAL

The time in seconds between retry attempts by a listener port to recover from a failure. The maximum number of retry attempts is defined by the MAX.RECOVERY.RETRIES custom property.

A failure can be one of two things:

- An unexpected error has occurred when a listener port tries to get a message from the JMS provider.
- The connection between the application server and the JMS provider has been lost, usually due to a network error.

Data type: Integer
Units: Seconds
Default: 60
Range: 1 through 2147483647

+++ end +++