How to read and write fields from the MQMD via WebSphere MQ Classes for JMS Version 7: direct, using JNDI and using WebSphere Application Server

IBM Techdoc: 7020570

http://www.ibm.com/support/docview.wss?rs=171&uid=swg27020570

Date last updated: 27-Oct-2010

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

+++ Objective +++

The objective of this techdoc is to show how to read and write the MQMD fields in a message, using the WebSphere MQ JMS Version 7 code. A portion of the needed information is located in different links of the MQ and WAS Information Centers. This article tries to put in a single place all the different pieces of the puzzle.

This document shows all the steps and excerpts of source code for the following scenarios:

1) Direct specification of Connection Factory and Destination (Not using JNDI)
2) Using JNDI (via file system context)
3) Using an MDB inside WebSphere Application Server (WAS) V7

+++ Introduction +++

MQ V7 provides the facility to read and write the MQMD fields from an MQ JMS application.

For more information, see the following references:

http://publib.boulder.ibm.com/infocenter/wmqv7/v7r0/index.jsp?topic=/com.ibm.mq.csqzaw.doc/jm41030_.htm
MQ Information Center V7
Reading and writing the message descriptor from a WebSphere MQ classes for JMS application

http://publib.boulder.ibm.com/infocenter/wmqv7/v7r0/index.jsp?topic=/com.ibm.mq.csqzaw.doc/jm41040_.htm
MQ Information Center V7
Destination object properties

http://publib.boulder.ibm.com/infocenter/wmqv7/v7r0/index.jsp?topic=/com.ibm.mq.csqzaw.doc/jm41050_.htm
MQ Information Center V7
Message object properties

http://publib.boulder.ibm.com/infocenter/wmqv7/v7r0/index.jsp?topic=/com.ibm.mq.csqzaw.doc/jm35150_.htm
MQ Information Center V7
Using the IBM JMS extensions

http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/cmm_customprops.html
WAS V7 Information Center > WebSphere MQ custom properties

http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/tmm_customproperties.html
WAS V7 Information Center > Configuring custom properties for the WebSphere MQ messaging provider

Between the above links, all the information on how to read/write MQMD fields using MQ JMS is provided, but the readers need to consult separate sections of the Information Centers, because they do not provide complete scenarios. Several MQ customers requested to have full scenarios that describe all the steps

The software used for the preparation of this document is:
- Windows XP using WebSphere MQ 7.0.1.3 with Java 1.6.
- Linux Intel x86-32 bit, SLES 11, using WebSphere MQ 7.0.1.3, WAS 7.0.13, Java 1.6.
- Linux PowerPC, SLES 9, using WebSphere MQ 7.0.1.1

For the testing with WAS, the following articles were referenced and the sample source code for the onMethod() of the MDB was used and expanded:

Using WebSphere MQ V7 as JMS Provider for WebSphere Application Server V7
http://www.ibm.com/support/docview.wss?rs=171&uid=swg27016505

Developing and testing an MDB using RAD 7.5, WebSphere Application Server V7 and MQ V7 as JMS Provider

http://www.ibm.com/support/docview.wss?rs=171&uid=swg27016507

To facilitate your testing, the 5 files with sample code mentioned in this article are provided in a zip file:

MQ-JMS-MDB.zip

Scenario 1:
    SimpleMQMDWrite.java
    SimpleMQMDRead.java
Scenario 2:
    JmsJndiConsumerMQMD.java
    JmsJndiProducerMQMD.java
Scenario 3:
    onMethod-mqmd.java

+++ DISCLAIMER

All source code and/or binaries attached to this document are referred to here as "the Program". IBM is not providing program services of any kind for the Program. IBM is providing the Program on an "AS IS" basis without warranty of any kind. IBM WILL NOT BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, INCIDENTAL, OR INDIRECT DAMAGES OR FOR ANY ECONOMIC CONSEQUENTIAL DAMAGES (INCLUDING LOST PROFITS OR SAVINGS), EVEN IF IBM, OR ITS RESELLER, HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

++
+++ Chapter 1: Direct specification of Connection Factory and Destination
+++ (Not using JNDI)
++

MQ V7 provides the following 2 JMS samples that do NOT use JNDI for the specification of the Connection Factory and of the Destination (such as Queue). These are the samples:

Unix:
/opt/mqm/samp/jms/samples/simple/SimpleMQMDWrite.java
/opt/mqm/samp/jms/samples/simple/SimpleMQMDRead.java

Windows:
C:\Program Files\IBM\WebSphere MQ\tools\jms\samples\simple\SimpleMQMDWrite.java
C:\Program Files\IBM\WebSphere MQ\tools\jms\samples\simple\SimpleMQMDRead.java

+ Writing into the MQMD attributes: SimpleMQMDWrite

It is worth noting the following statements from the sample.
You must cast the "destination" object to a "JmsDestination" and use the constant to enable to write into the MQMD fields.

```
    // Enable write of MQMD fields. See documentation for further details.
    ((JmsDestination)
destination).setBooleanProperty(WMQConstants.WMQ_MQMD_WRITE_ENABLED, true);
```

In this sample, the messageID is going to be customized and this is the property to be modified:

```
   // Write to MQMD.MsgId via JMS_IBM_MQMD_MSGID message property
    message.setObjectProperty(WMQConstants.JMS_IBM_MQMD_MSGID,
customMessageId);
```

It is recommended that you do not work directory with the samples provided by MQ. Instead, you should copy the sample java files into a working directory outside the MQ code directory (outside /opt/mqm, for example).

Proceed to modify the source code to reflect the names of your test queue manager and queue:
- Change the name of the queue manager (QM1) and/or the queue (Q1).
    cf.setStringProperty(WMQConstants.WMQ_QUEUE_MANAGER, "QM1");

…

destination = session.createQueue("queue:///Q1");

Note: To simplify the compilation and runtime of the code, I commented out the statement:

package simple;

In Unix, set the MQ JMS environment and compile the code:

javac SimpleMQMDWrite

Then run it. Notice the JMS_IBM_MQMD_MsgId (it is the customized one).
Notice also that the JMSDestination is marked as "mdWriteEnabled=true".

java SimpleMQMDWrite

```
  JMSMessage class: jms_text
  JMSType:          null
  JMSDeliveryMode:  2
  JMSExpiration:    0
  JMSPriority:      4
  JMSMessageID:     ID:0102030405060708010203040506070801020304050607 08
  JMSTimestamp:     1255027961111
  JMSCorrelationID: null
  JMSDestination:   queue:///Q1?mdWriteEnabled=true
  JMSReplyTo:       null
  JMSRedelivered:   false
    JMSXAppID: WebSphere MQ Client for Java
    JMSXDeliveryCount: 0
    JMSXUserID: rivera
    JMS_IBM_MQMD_MsgId: 0102030405060708010203040506070801020304050607 08
    JMS_IBM_PutApplType: 28
    JMS_IBM_PutDate: 20091008
    JMS_IBM_PutTime: 18524115
SimpleMQMDWrite: Your lucky number today is 38
SUCCESS
```

+++ Reading the MQMD attributes: SimpleMQMDRead

It is worth noting the following statements from the source code, which will enable the code to read the MQMD fields:

```
    // Enable read of MQMD fields. See documentation for further details.
     ((JmsDestination)
destination).setBooleanProperty(WMQConstants.WMQ_MQMD_READ_ENABLED, true);
```

Follow the same recommendations from the above section.

javac SimpleMQMDRead

Then run it. Notice the JMS_IBM_MQMD_MsgId (it is the customized one):
Notice also that the JMSDestination is marked as "mdWriteEnabled=true" (I thought that it would say: mqReadEanbled=true).

java SimpleMQMDRead

```
Received message:
  JMSMessage class: jms_text
  JMSType:          null
  JMSDeliveryMode:  2
  JMSExpiration:    0
  JMSPriority:      4
  JMSMessageID:     ID:0102030405060708010203040506070801020304050607 08
  JMSTimestamp:     1255027961111
  JMSCorrelationID: null
  JMSDestination:   queue:///Q1?mdWriteEnabled=true
  JMSReplyTo:       null
  JMSRedelivered:   false
    JMSXAppID: WebSphere MQ Client for Java
    JMSXDeliveryCount: 1
    JMSXUserID: rivera
    JMS_IBM_Character_Set: UTF-8
    JMS_IBM_Encoding: 273
    JMS_IBM_Format: MQSTR
    JMS_IBM_MQMD_AccountingToken:
0000000000000000000000000000000000000000000000000000000000000000
    JMS_IBM_MQMD_ApplIdentityData:
    JMS_IBM_MQMD_ApplOriginData:
    JMS_IBM_MQMD_BackoutCount: 0
    JMS_IBM_MQMD_CodedCharSetId: 1208
    JMS_IBM_MQMD_CorrelId: 000000000000000000000000000000000000000000000000
    JMS_IBM_MQMD_Encoding: 273
    JMS_IBM_MQMD_Expiry: -1
    JMS_IBM_MQMD_Feedback: 0
    JMS_IBM_MQMD_Format: MQHRF2
    JMS_IBM_MQMD_GroupId: 000000000000000000000000000000000000000000000000
    JMS_IBM_MQMD_MsgFlags: 0
    JMS_IBM_MQMD_MsgId: 0102030405060708010203040506070801020304050607 08
    JMS_IBM_MQMD_MsgSeqNumber: 1
    JMS_IBM_MQMD_MsgType: 8
    JMS_IBM_MQMD_Offset: 0
    JMS_IBM_MQMD_OriginalLength: -1
    JMS_IBM_MQMD_Persistence: 1
    JMS_IBM_MQMD_Priority: 4
    JMS_IBM_MQMD_PutApplName: WebSphere MQ Client for Java
    JMS_IBM_MQMD_PutApplType: 28
    JMS_IBM_MQMD_PutDate: 20091008
    JMS_IBM_MQMD_PutTime: 18524115
    JMS_IBM_MQMD_ReplyToQ:
    JMS_IBM_MQMD_ReplyToQMgr: QM_VER
    JMS_IBM_MQMD_Report: 0
    JMS_IBM_MQMD_UserIdentifier: rivera
    JMS_IBM_MsgType: 8
    JMS_IBM_PutApplType: 28
    JMS_IBM_PutDate: 20091008
    JMS_IBM_PutTime: 18524115
SimpleMQMDWrite: Your lucky number today is 38
SUCCESS
```

++
+++ Chapter 2: Using JNDI (via file system context)
++

The existing sample code provided with MQ was used as the base for new samples:

Unix:
/opt/mqm/samp/jms/samples/JmsJndiConsumer.java
/opt/mqm/samp/jms/samples/JmsJndiProducer.java

Windows:
C:\Program Files\IBM\WebSphere MQ\tools\jms\samples\JmsJndiConsumer.java
C:\Program Files\IBM\WebSphere MQ\tools\jms\samples\JmsJndiProducer.java

The following examples are based on the above existing sample code. The new examples were modified to add the MQMD write/read capability.

JmsJndiConsumerMQMD.java
JmsJndiProducerMQMD.java

In the new code, you have to perform the code changes mentioned in the previous scenario. Also, because the customized property is "JMS_IBM_MQMD_ApplIdentityData", it is also required that you setup the proper context.

+ To write (JmsJndiProducerMQMD.java):

```
import com.ibm.msg.client.wmq.WMQConstants;
…
     // Enable MQMD write
      ((JmsDestination)
destination).setBooleanProperty(WMQConstants.WMQ_MQMD_WRITE_ENABLED, true);
…
     // Set a message context for this MD field
      ((JmsDestination)
destination).setIntProperty(WMQConstants.WMQ_MQMD_MESSAGE_CONTEXT,
               WMQConstants.WMQ_MDCTX_SET_IDENTITY_CONTEXT);
...

     // On the message, set property to provide custom ApplIdentityData
     message.setStringProperty("JMS_IBM_MQMD_ApplIdentityData",
"TestValueApplIdentityData");
```

+ To read (JmsJndiConsumerMQMD.java):

```
import com.ibm.msg.client.wmq.WMQConstants;
…
     // Enable MQMD read
      ((JmsDestination)
destination).setBooleanProperty(WMQConstants.WMQ_MQMD_READ_ENABLED, true);
…
     // Set a message context if applicable for this MD field
      ((JmsDestination)
destination).setIntProperty(WMQConstants.WMQ_MQMD_MESSAGE_CONTEXT,
               WMQConstants.WMQ_MDCTX_SET_IDENTITY_CONTEXT);
…
     // Receive the message
     Message message = consumer.receive(timeout);
     if (message != null) {
       System.out.println("Received message:\n" + message);
       String property1 =
message.getStringProperty("JMS_IBM_MQMD_ApplIdentityData");
       System.out.println("MQMD - Retrieved ApplicationIdentityData = " + property1);
     }
```

But the code change is NOT enough.

At this point, if you run the producer and then run the consumer, you will notice that the property in question will NOT have the value.
This is due to the values for MQREAD and MQWRITE from the JNDI Destination object for a Queue are set to the default 'NO', which does NOT allow to read/write the MQMD:

The following output is shown via the JMSAdmin tool and indicates the default values:

```
InitCtx> display q(Q1)
     CCSID(437)
     DESCRIPTION(Queue-with-QMgrName)
     ENCODING(NATIVE)
     EXPIRY(APP)
     FAILIFQUIESCE(YES)
     MDMSGCTX(DEFAULT)
     MDREAD(NO)
     MDWRITE(NO)
     MSGBODY(UNSPECIFIED)
     PERSISTENCE(APP)
     PRIORITY(APP)
     PUTASYNCALLOWED(AS_DEST)
     QMANAGER(QM_ANGELITO)
     QUEUE(Q1)
```

```
    READAHEADALLOWED(AS_DEST)
    READAHEADCLOSEPOLICY(DELIVER_ALL)
    REPLYTOSTYLE(DEFAULT)
    TARGCLIENT(JMS)
    VERSION(7)
```

You need to use the JMSAdmin tool or the MQ Explorer to modify the JNDI Destination object for the Queue to change the values to **YES**:

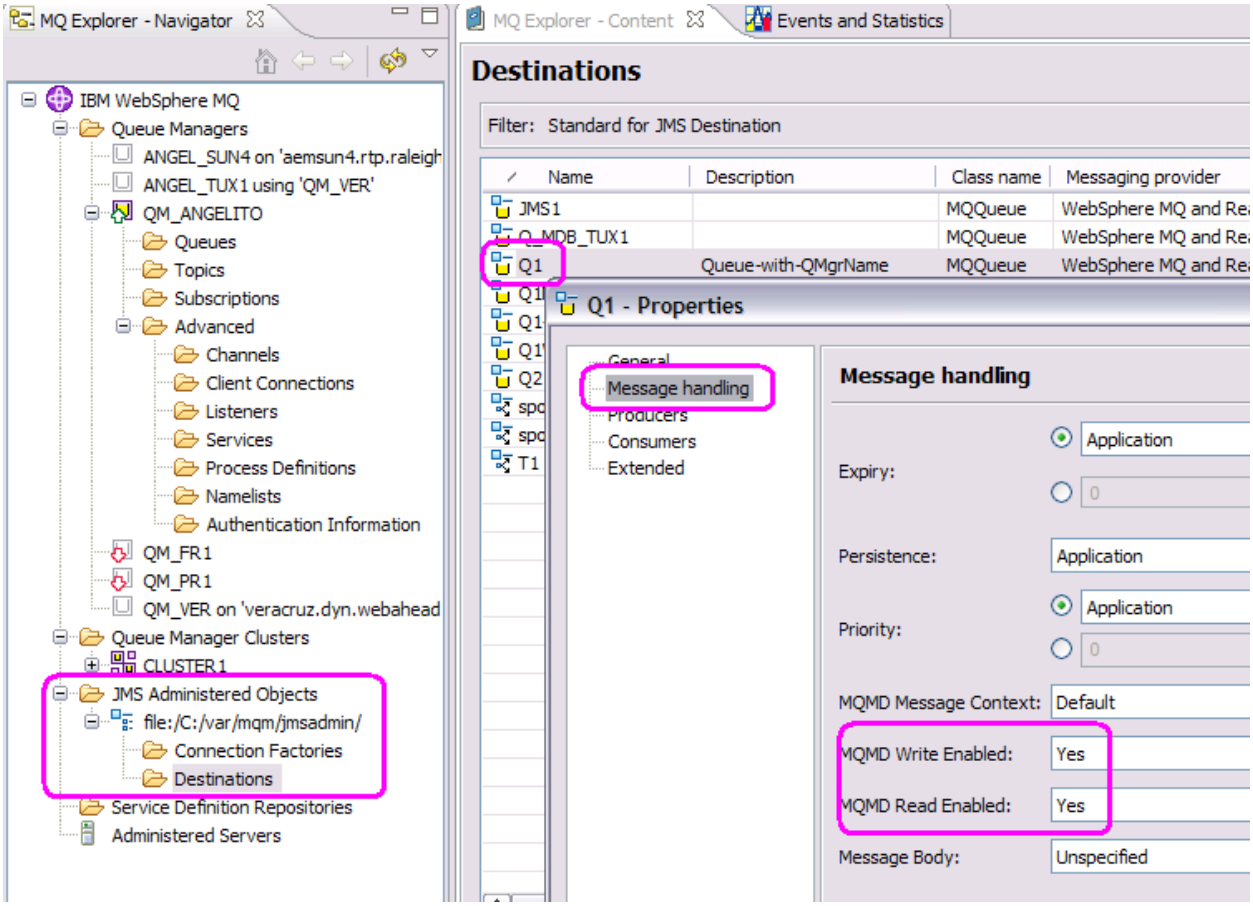
+ Using JMSAdmin

```
InitCtx> alter q(Q1) MDREAD(YES) MDWRITE(YES)

InitCtx> display q(Q1)

    CCSID(437)
...
    MDMSGCTX(DEFAULT)
    MDREAD(YES)
    MDWRITE(YES)
    MSGBODY(UNSPECIFIED)
...
    TARGCLIENT(JMS)
    VERSION(7)
```

+ Using the MQ Explorer:

From the left panel, click on "JMS Administered Objects" then expand "Destinations".
On the right panel, select the desired Queue, and right click. Then select "Properties"
Select the tab "Message Handling".
Change the values from NO to YES for:
MQMD Write Enabled
MQMD Read Enabled
Click OK

Notes:
- The connection factory object "CF1Client" was created for this example.
- The location of the JNDI file ".bindings" (file system context) for this example is stored under:
C:/var/mqm/jmsadmin



+ Running the samples

Notice the value for the customized property. This is the value placed by the write sample:

JMS_IBM_MQMD_ApplIdentityData: TestValueApplIdentityData

Run the Producer of the message.
C:\angel\coding\MQ\jms\mqv7-jndi>
java JmsJndiProducerMQMD -i file:/C:/var/mqm/jmsadmin -c CF1Client -d Q1

```
Initial context found!
Sent message:
  JMSMessage class: jms_text
  JMSType:          null
  JMSDeliveryMode:  2
  JMSExpiration:    0
  JMSPriority:      4
  JMSMessageID:     ID:414d5120514d5f414e47454c49544f20c6d9f74c20004d03
  JMSTimestamp:     1291314365671
  JMSCorrelationID: null
  JMSDestination:
queue://QM_ANGELITO/Q1?mdReadEnabled=true&mdWriteEnabled=true&destDescription=
Queue-with-QMgrName&CCSID=437&mdMessageContext=1
```

```
  JMSReplyTo:       null
  JMSRedelivered:   false
    JMSXAppID: WebSphere MQ Client for Java
    JMSXDeliveryCount: 0
    JMS_IBM_MQMD_AccountingToken:
000000000000000000000000000000000000000000000000
00000000000000000
    JMS_IBM_MQMD_ApplIdentityData: TestValueApplIdentityData
    JMS_IBM_MQMD_ApplOriginData:
    JMS_IBM_MQMD_BackoutCount: 0
    JMS_IBM_MQMD_CodedCharSetId: 1208
    JMS_IBM_MQMD_CorrelId: 000000000000000000000000000000000000000000000000
    JMS_IBM_MQMD_Encoding: 273
    JMS_IBM_MQMD_Expiry: -1
    JMS_IBM_MQMD_Feedback: 0
    JMS_IBM_MQMD_Format: MQHRF2
    JMS_IBM_MQMD_GroupId: 000000000000000000000000000000000000000000000000
    JMS_IBM_MQMD_MsgFlags: 0
    JMS_IBM_MQMD_MsgId: 414D5120514D5F414E47454C49544F20C6D9F74C20004D03
    JMS_IBM_MQMD_MsgSeqNumber: 1
    JMS_IBM_MQMD_MsgType: 8
    JMS_IBM_MQMD_Offset: 0
    JMS_IBM_MQMD_OriginalLength: -1
    JMS_IBM_MQMD_Persistence: 1
    JMS_IBM_MQMD_Priority: 4
    JMS_IBM_MQMD_PutApplName: WebSphere MQ Client for Java
    JMS_IBM_MQMD_PutApplType: 28
    JMS_IBM_MQMD_PutDate: 20101202
    JMS_IBM_MQMD_PutTime: 18260568
    JMS_IBM_MQMD_ReplyToQ: null
    JMS_IBM_MQMD_ReplyToQMgr: null
    JMS_IBM_MQMD_Report: 0
    JMS_IBM_MQMD_UserIdentifier: null
    JMS_IBM_PutApplType: 28
    JMS_IBM_PutDate: 20101202
    JMS_IBM_PutTime: 18260568
JmsJndiProducerMQMD: Your lucky number today is 640
SUCCESS
```

Run the Consumer:

C:\angel\coding\MQ\jms\mqv7-jndi>
java JmsJndiConsumerMQMD -i file:/C:/var/mqm/jmsadmin -c CF1Client -d Q1

```
Initial context found!
Received message:

  JMSMessage class: jms_text
  JMSType:          null
  JMSDeliveryMode:  2
  JMSExpiration:    0
  JMSPriority:      4
  JMSMessageID:     ID:414d5120514d5f414e47454c49544f20c6d9f74c20004d03
  JMSTimestamp:     1291314365671
  JMSCorrelationID: null
  JMSDestination:
queue://QM_ANGELITO/Q1?mdReadEnabled=true&mdWriteEnabled=true&destDescription=
Queue-with-QMgrName&CCSID=437&mdMessageContext=1
  JMSReplyTo:       null
  JMSRedelivered:   false
```

```
    JMSXAppID: WebSphere MQ Client for Java
    JMSXDeliveryCount: 1
    JMSXUserID:
    JMS_IBM_Character_Set: UTF-8
    JMS_IBM_Encoding: 273
    JMS_IBM_Format: MQSTR
    JMS_IBM_MQMD_AccountingToken:
0000000000000000000000000000000000000000000000000000000000000000
    JMS_IBM_MQMD_ApplIdentityData: TestValueApplIdentityData
    JMS_IBM_MQMD_ApplOriginData:
    JMS_IBM_MQMD_BackoutCount: 0
    JMS_IBM_MQMD_CodedCharSetId: 1208
    JMS_IBM_MQMD_CorrelId: 000000000000000000000000000000000000000000000000
    JMS_IBM_MQMD_Encoding: 273
    JMS_IBM_MQMD_Expiry: -1
    JMS_IBM_MQMD_Feedback: 0
    JMS_IBM_MQMD_Format: MQHRF2
    JMS_IBM_MQMD_GroupId: 000000000000000000000000000000000000000000000000
    JMS_IBM_MQMD_MsgFlags: 0
    JMS_IBM_MQMD_MsgId: 414D5120514D5F414E47454C49544F20C6D9F74C20004D03
    JMS_IBM_MQMD_MsgSeqNumber: 1
    JMS_IBM_MQMD_MsgType: 8
    JMS_IBM_MQMD_Offset: 0
    JMS_IBM_MQMD_OriginalLength: -1
    JMS_IBM_MQMD_Persistence: 1
    JMS_IBM_MQMD_Priority: 4
    JMS_IBM_MQMD_PutApplName: WebSphere MQ Client for Java
    JMS_IBM_MQMD_PutApplType: 28
    JMS_IBM_MQMD_PutDate: 20101202
    JMS_IBM_MQMD_PutTime: 18260568
    JMS_IBM_MQMD_ReplyToQ:
    JMS_IBM_MQMD_ReplyToQMgr: QM_ANGELITO
    JMS_IBM_MQMD_Report: 0
    JMS_IBM_MQMD_UserIdentifier:
    JMS_IBM_MsgType: 8
    JMS_IBM_PutApplType: 28
    JMS_IBM_PutDate: 20101202
    JMS_IBM_PutTime: 18260568
JmsJndiProducerMQMD: Your lucky number today is 640
MQMD - Retrieved ApplicationIdentityData = TestValueApplIdentityData
SUCCESS
```

++
+++ Chapter 3: Using an MDB inside WebSphere Application Server (WAS) V7
++

+ MDB code

This scenario uses as base, the MDB that is developed by following the instructions from this article:

Developing and testing an MDB using RAD 7.5, WebSphere Application Server V7 and MQ V7 as JMS Provider
http://www.ibm.com/support/docview.wss?rs=171&uid=swg27016507

Modify the onMessage( ) code of the MDB as follows, to add the processing for reading the MQMD property: JMS_IBM_MQMD_ApplIdentityData

```
public void onMessage(javax.jms.Message msg) {
  try {
       if (msg instanceof javax.jms.TextMessage) {
          System.out.println("+++ SAMPLE MDB: Text Message => " +
((javax.jms.TextMessage)msg).getText());
          String property1 = msg.getStringProperty("JMS_IBM_MQMD_ApplIdentityData");
          System.out.println("              Property JMS_IBM_MQMD_ApplIdentityData => "
+ property1);
     }
…
```

Then you have to export the EJB into an EAR

+ Configure the baseline setup

It is recommended that you first deploy and test the base MDB mentioned in the following article. If you encounter problems during the deployment or at runtime, you need to address them before proceeding.

Using WebSphere MQ V7 as JMS Provider for WebSphere Application Server V7
http://www.ibm.com/support/docview.wss?rs=171&uid=swg27016505

Once you have an MDB that is up and running and which has been tested, then you can proceed to deploy the new MDB that has the code changes for reading the MQMD.

+ Modifications to the WAS JMS Resources

Assuming that you have now a good and tested baseline of the basic MDB.
Now, you will need to define some additional properties to the JMS Queue Destination object. The details of what is added is mentioned in this link:

http://publib.boulder.ibm.com/infocenter/wmqv7/v7r0/index.jsp?topic=/com.ibm.mq.csqzaw.doc/jm41040_.htm
Destination object properties

The details on how to make the modifications are shown in the following screenshots.

You will need to create 3 custom properties:

Name	Value	Description
MDR	YES	WMQ_MQMD_READ_ENABLED
MDW	YES	WMQ_MQMD_WRITE_ENABLED
MDCTX	SET_IDENTITY_CONTEXT	WMQ_MQMD_MESSAGE_CONTEXT

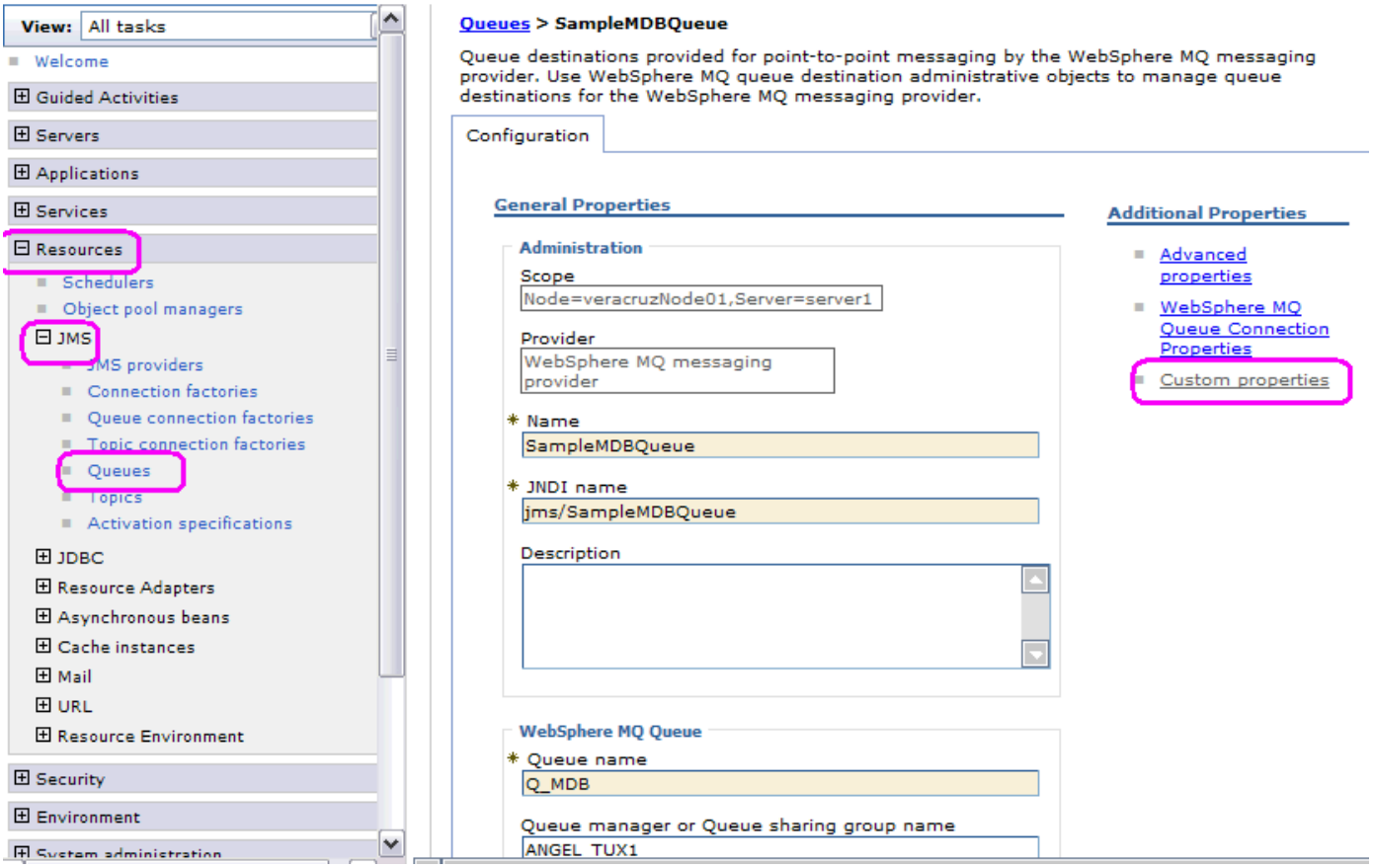
Technically speaking, because this sample MDB is only reading a message, only the MDR and MDCTX properties are needed. However, because you may later on have an EJB that writes into the MQMD of a message, it is a good idea to include MDW at this time.

An additional reason is that any time that you add a new JMS resource or modify a JMS resource, you will need to reboot the WAS server in order to make effective the changes. Thus, to minimize the number of times that you need to restart the server, it is best to make all the possible JMS changes at one time.

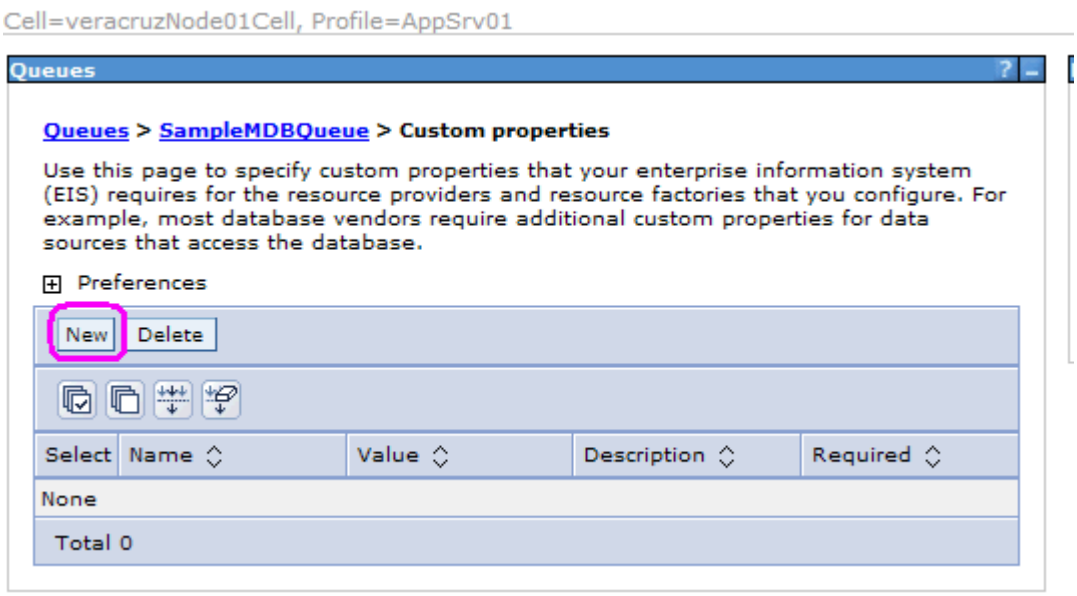
This document shows all the steps for creating the first custom property. Then you can repeat the steps for the other properties.

From the WAS Administrative Console, select Resources > JMS > Queues
And select the desired Queue Destination.

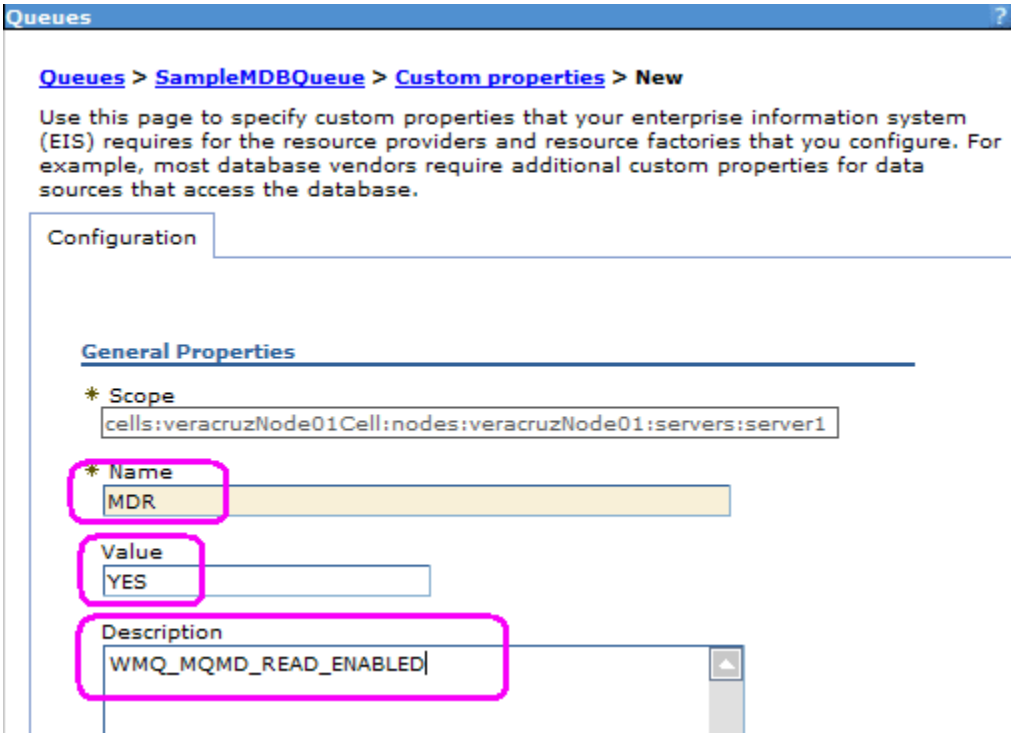
Then click on "Custom properties".



You will see the following dialog.

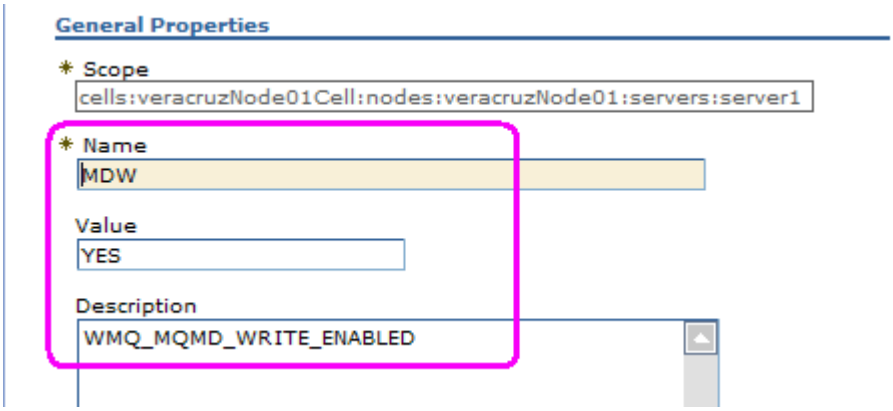


Click on New.
Enter the name and value for the first property:
Name: MDR
Value: YES
Description: WMQ_MQMD_READ_ENABLED



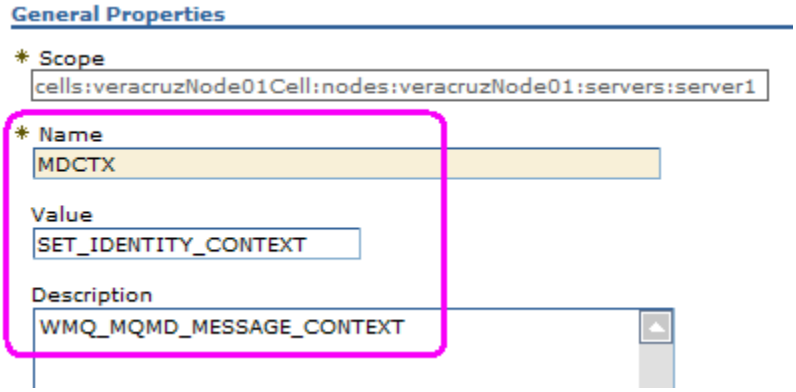
Click OK

Proceed to create the next property:



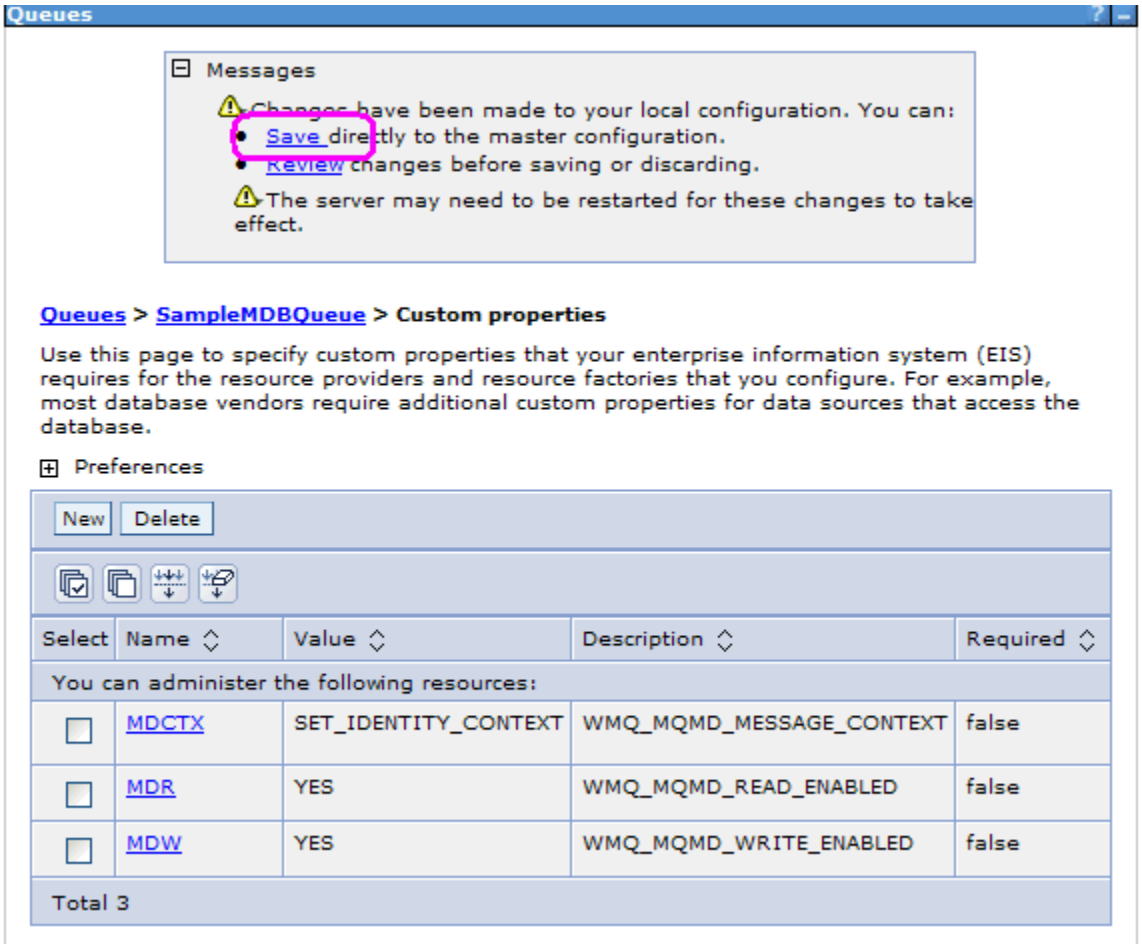
Click OK.

Proceed to create the last property:



Click OK

You should see the following 3 custom properties.
Click on Save.



You must restart the WAS Server, because changes to the JMS Resources require it.

Then start the Listener Port and the MDB

There are 2 test cases:

a) Using a non-JMS application to put a message (no value in the property JMS_IBM_MQMD_ApplIdentityData)

In this scenario, the queue manager (ANGEL_TUX1) is running in a Linux host called "aemtux1" and the queue Q_MDB is the one that the Listener Port in WAS is listening to.

The MQ sample "amqsput" is used to put a message into the queue:

aemtux1:/rivera 1006% amqsput Q_MDB ANGEL_TUX1
Sample AMQSPUT0 start
target queue is Q_MDB
test3

The Listener Port in WAS will deliver the message to the modified MDB, which will then issue 2 print statements in the SystemOut.log file as follows:

[12/2/10 14:47:10:248 EST] 0000002a SystemOut O +++ SAMPLE MDB: Text Message => test3
[12/2/10 14:47:10:249 EST] 0000002a SystemOut O Property JMS_IBM_MQMD_ApplIdentityData =>

Notice that the message data is: test3
However, notice that there is no value for the property, because amqsput does not write into that field:
JMS_IBM_MQMD_ApplIdentityData

b) Using a JMS application that writes into the MQMD property:
JMS_IBM_MQMD_ApplIdentityData

The JMS application "JmsJndiProducerMQMD" mentioned in Chapter 2 is used for this purpose. It is going to be run from Windows and the queue manager is in Linux.
The connection factory CF1TUX1Client has the proper connection details and the Queue Destination Q_MDB_TUX1 has the value YES for the fields MQREAD and MQWRITE as explained in Chapter 2.

C:\angel\coding\MQ\jms\mqv7-jndi>
java JmsJndiProducerMQMD -i file:/C:/var/mqm/jmsadmin -c CF1TUX1Client -d Q_MDB_TUX1

```
Initial context found!
Sent message:

  JMSMessage class: jms_text
  JMSType:          null
  JMSDeliveryMode:  2
  JMSExpiration:    0
  JMSPriority:      4
  JMSMessageID:     ID:414d5120414e47454c5f5455583120204cdc1ac220005b02
  JMSTimestamp:     1291318778531
  JMSCorrelationID: null
  JMSDestination:
queue://ANGEL_TUX1/Q_MDB?mdReadEnabled=true&mdWriteEnabled=true&mdMessageConte
xt=1
```

```
  JMSReplyTo:       null
  JMSRedelivered:   false
    JMSXAppID: WebSphere MQ Client for Java
    JMSXDeliveryCount: 0
    JMS_IBM_MQMD_AccountingToken:
000000000000000000000000000000000000000000000000
000000000000000000
    JMS_IBM_MQMD_ApplIdentityData: TestValueApplIdentityData
    JMS_IBM_MQMD_ApplOriginData:
    JMS_IBM_MQMD_BackoutCount: 0
    JMS_IBM_MQMD_CodedCharSetId: 1208
    JMS_IBM_MQMD_CorrelId: 000000000000000000000000000000000000000000000000
    JMS_IBM_MQMD_Encoding: 273
    JMS_IBM_MQMD_Expiry: -1
    JMS_IBM_MQMD_Feedback: 0
    JMS_IBM_MQMD_Format: MQHRF2
    JMS_IBM_MQMD_GroupId: 000000000000000000000000000000000000000000000000
    JMS_IBM_MQMD_MsgFlags: 0
    JMS_IBM_MQMD_MsgId: 414D5120414E47454C5F5455583120204CDC1AC220005B02
    JMS_IBM_MQMD_MsgSeqNumber: 1
    JMS_IBM_MQMD_MsgType: 8
    JMS_IBM_MQMD_Offset: 0
    JMS_IBM_MQMD_OriginalLength: -1
    JMS_IBM_MQMD_Persistence: 1
    JMS_IBM_MQMD_Priority: 4
    JMS_IBM_MQMD_PutApplName: WebSphere MQ Client for Java
    JMS_IBM_MQMD_PutApplType: 28
    JMS_IBM_MQMD_PutDate: 20101202
    JMS_IBM_MQMD_PutTime: 19393855
    JMS_IBM_MQMD_ReplyToQ: null
    JMS_IBM_MQMD_ReplyToQMgr: null
    JMS_IBM_MQMD_Report: 0
    JMS_IBM_MQMD_UserIdentifier: null
    JMS_IBM_PutApplType: 28
    JMS_IBM_PutDate: 20101202
    JMS_IBM_PutTime: 19393855
JmsJndiProducerMQMD: Your lucky number today is 500
SUCCESS
```

The bottom of SystemOut.log will show:

```
[12/2/10 14:47:17:931 EST] 0000002a SystemOut     O +++ SAMPLE MDB: Text
Message => JmsJndiProducerMQMD: Your lucky number today is 500
[12/2/10 14:47:17:932 EST] 0000002a SystemOut     O                 Property
JMS_IBM_MQMD_ApplIdentityData => TestValueApplIdentityData
```

Notice that the value for the MQMD field JMS_IBM_MQMD_ApplIdentityData is:
TestValueApplIdentityData

+++ end +++