

IBM MQ Resource Adapter

**IBM MQ Java EE Connector Architecture MDB
Verification Test Program (mdbfvt)**

**Ver 1.2
9 Oct, 2017**

**S.N.Software Inc.
e-mail : support@pulsarintegration.com**

Program Version 1.2.0.0

Tested versions of this program

· WebSphere Application Server 9.0.0.3(MQ RA 9.0.0.0) / IBM MQ 9.0.0.0 / IBM Java 8.0.4.6 / Windows 10 Pro

* This program may be executable in other WebSphere Application Server versions following Java EE 7.0 compliance.

Table of Contents

About this Program.....	4
1. Set Up the Environment for This Program.....	5
Deploying mdbfvt.....	5
Set JNI Libraries Location.....	5
User for This Program.....	5
Other Considerations.....	5
2. Preparation for Testing.....	7
Creating a Connection Factory.....	7
Creating a Queue.....	7
Creating a Activation Specification.....	7
Starting the Program.....	7
3. How to use this program.....	8
Receiving a TextMessage.....	8
4. Receiving Non-Text Type JMS Message.....	12
Receiving a MapMessage.....	12
Receiving an ObectMessage.....	14
Receiving a StreamMessage.....	15
Receiving a ByteMessage.....	16
Conclusion	19

About this Program

This program is created to verify / confirm the inbound function(MDB) / usage of WebSphere MQ / IBM MQ and its provided MQ resource adapter. (For outbound function, please use mqfvt verification program.) This program can perform the following verification.

EJB3.1(local) Container Managed Transaction, Activation specifications, receiving JMS Message Type: TextMessage/MapMessage/ObjectMessage/StreamMessage/ByteMessage

This document does not describe the details of IBM MQ itself. Please refer to the product manual as necessary.

Product manuals of all versions can be found at the following URL:

IBM MQ and WebSphere MQ

<http://www-01.ibm.com/software/integration/wmq/library/index.html>

MQRA verification program(mqfvt) is also used for the test using this program. Please refer to the document “IBM MQ Java EE Connector Architecture Verification Test Program(mqfvt)” for detail on them.

1. Set Up the Environment for This Program

As a prerequisite to using mdbfvt, it is necessary that the MQ server of IBM MQ 7.0.1 or higher is installed locally or remotely, and the version of WebSphere Application Server conforming to Java EE 7.0 is available.

Deploying mdbfvt

By installing MDBFVTEJBEAR.ear on your WebSphere Application Server, you can use mdbfvt.

You must have created an activation specification to use before installing the EAR. If it is not created, create it referring to "2. Preparation for Testing".

Here is an example for IBM WebSphere Application Server 9.0 (Traditional).

In the Admin Console, select "Applications" - "New Application" - "New Enterprise Application", then open the "Preparing for Application Installation" screen and specify the location of the MDBFVTEJBEAR.ear file. All except the below will complete the installation by default.

- Check "Deploy enterprise bean" in "Select installation option" screen.
- In "Binding listeners for message-driven beans", check "activation specification" and specify "target resource JNDI name" (JNDI name of activation specification).

Set JNI Libraries Location

When binding mode is used for connection to the queue manager, it may be necessary to explicitly specify the location of the JNI library (mqjbnd.dll, libmqjbnd.so, etc.) depending on the environment. In WebSphere Application Server 9.0 (Traditional), select "Resource" - "JMS" - "JMS provider", then select "WebSphere MQ messaging provider" within the appropriate scope and set the JNI library path according to your MQ environment to "Native library Path" of "General properties".

e.g.

C:\Program Files\IBM\MQ\java\lib64
/opt/mqm/java/lib
/usr/mqm/java/lib

User for This Program

The execution user of the application server may require appropriate access authority to be set in the queue manager. See the IBM MQ documentation for details.

Other Considerations

* On some WAS MQ administration screen, it may be necessary to set the path to the JNI library in LIBPATH (AIX) or LD_LIBRARY_PATH. It can be set in the following file.

<WAS Install Directory>\bin\setupCmdLine.bat(Windows)

<WAS Install Directory>/bin/setupCmdLine.sh(Unix)

* In some Windows environments, you may need to set <MQ Install Directory>\bin64 in the PATH environment variable. It can be set with setupCmdLine.bat(sh).

2. Preparation for Testing

To run the program, define the connection factory, activation specification and queue on the application server.

The following is an example for IBM WebSphere Application Server 9.0 (Traditional).

Creating a Connection Factory

In the management console, select "Resource" - "JMS" - "Connection Factory" to open the "Connection Factory" screen and then click the "Create New" button. Then select "WebSphere MQ Messaging Provider" and specify "name" and "JNDI name" on the "Create IBM MQ JMS Resource" screen. A connection factory name specified in an application is "JNDI name" to be entered here. (Since JNDI ENC name can not be used as this program does not define resource reference. So, it is necessary to specify JNDI simple name or JNDI fully qualified name.) After that specify the appropriate parameters and complete this procedure.

Creating a Queue

In the management console, select "Resource" - "JMS" - "Queue" to open the "Queue" screen and then click the "Create New" button. Select "WebSphere MQ messaging provider" and specify "name", "JNDI name", "queue name" (real queue name). (If you do not specify "queue manager or queue share group name", you can use it generically.) The queue name you specify from an application is "JNDI name" you enter here. After that, specify the appropriate parameters and complete this procedure.

Creating a Activation Specification

In the management console, select "Resource" - "JMS" - "Activation Specification" to open the "Activation Specification" screen and then click the "Create New" button. Select "WebSphere MQ messaging provider" and specify "name", "JNDI name", "Destination JNDI name" (JNDI name of the queue defined in the previous section). After that, specify the appropriate parameters and complete this procedure.

Starting the Program

In the Admin Console, select Applications - Application Type - WebSphere Enterprise Application to open "Enterprise Application", and then select(check) "MDBFVTEJBEAR" and click "Start".

3. How to use this program

This section will describe how to use this program as an example of receiving TextMessage.

Receiving a TextMessage

Send a JMS message by the MQRA verification program.

Received messages can be forwarded by specifying "queue URI(Uniform Resource Identifier)" in JMSReplyTo and a connection factory with prefix "CF:" in JMSType.

e.g.

JMSReplyTo: queue://mq90b/JMSQueue

* This value is passed to the argument of Session.createProducer(). It is unnecessary to define the queue specified on WAS. When getting a message by MQRA verification program, queue definition on WAS is required.

Set as follows and click "Submit".

IBM MQ Java EE Connector Architecture Veri

☒ : IBM MQ classes for JMS

☐ : IBM MQ classes for Java

☒ : EJB3.1 (Container Managed Transaction) ☐ : Web Container

EJB Transaction Attribute:

☐ : JMS Session transacted

☐ : JTA UserTransaction

☐ : Commit

☐ : Rollback

Sleep (before session close): sec

Connection Factory:

Destination:

☒ : put message

☐ : get message

JMS Message Type:

String:

Integer:

JMS header fields

JMSDeliveryMode:

JMSExpiration: ms

JMSPriority:

JMSCorrelationID: ID:<Hex> .. or String

JMSReplyTo:

JMSType:

(Specified parameters)

IBM MQ Classes for JMS : check

Connection Factory : jms/mq90BindCF *JNDI Simple Name for Connection Factory.

Destination : jms/SampleMDBQueue *JNDI Name for Queue.

put message : check

JMS Message Type: TextMessage

String : "MDB test message" *message string for PUT

After clicking “submit”, the following will be displayed.

Send message:

JMSMessage class: jms_text

JMSType: CF:jms/mq90bBindCF
JMSDeliveryMode: 1
JMSDeliveryDelay: 0
JMSDeliveryTime: 1507516719497
JMSExpiration: 0
JMSPriority: 0
JMSMessageID: ID:414d51206d71393020202020202025b9da59220c0503
JMSTimestamp: 1507516719497
JMSCorrelationID: null
JMSDestination: queue:///SampleMDBQueue
JMSReplyTo: queue://mq90b/JMSQueue
JMSRedelivered: false
JMSXAppID: Server\java\8.0\bin\java.exe
JMSXDeliveryCount: 0
JMSXUserID: okada
JMS_IBM_PutApplType: 11
JMS_IBM_PutDate: 20171009
JMS_IBM_PutTime: 02383949

MDB test message

SUCCESS

Confirmation of SystemOut.log:

Confirm with SystemOut.log that MDB has processed the message.

[17/10/09 11:38:39:515 JST] 00000115 SystemOut O +++ MDBFVTEJB: onMessage() start, this=-491445185

[17/10/09 11:38:39:515 JST] 00000115 SystemOut O +++ MDBFVTEJB: getJMSType=CF:jms/mq90bBindCF -491445185

[17/10/09 11:38:39:516 JST] 00000115 SystemOut O +++ MDBFVTEJB: Connection Factory for Reply jms/mq90bBindCF -491445185

[17/10/09 11:38:39:534 JST] 00000115 SystemOut O +++ MDBFVTEJB: JMS Message =>
JMSMessage class: jms_text
JMSType: CF:jms/mq90bBindCF
JMSDeliveryMode: 1
JMSDeliveryDelay: 0
JMSDeliveryTime: 0
JMSExpiration: 0

```

JMSPriority: 0
JMSMessageID: ID:414d51206d713930202020202020202025b9da59220c0503
JMSTimestamp: 1507516719497
JMSCorrelationID: null
JMSDestination: queue:///SampleMDBQueue
JMSReplyTo: queue://mq90b/JMSQueue
JMSRedelivered: false
JMSXAppID: Server\java\8.0\bin\java.exe
JMSXDeliveryCount: 1
JMSXUserID: okada
JMS_IBM_Character_Set: UTF-8
JMS_IBM_Encoding: 273
JMS_IBM_Format: MQSTR
JMS_IBM_MsgType: 1
JMS_IBM_PutApplType: 11
JMS_IBM_PutDate: 20171009
JMS_IBM_PutTime: 02383949
MDB test message

[17/10/09 11:38:39:535 JST] 00000115 SystemOut O +++ MDBFVTEJB: getJMSReplyTo =>
queue://mq90b/JMSQueue

[17/10/09 11:38:39:535 JST] 00000115 SystemOut O +++ MDBFVTEJB: Text Message =>
MDB test message

[17/10/09 11:38:39:535 JST] 00000115 SystemOut O +++ MDBFVTEJB: reply text => MDB
test message processed by: -491445185

[17/10/09 11:38:39:535 JST] 00000115 SystemOut O +++ MDBFVTEJB: sendReply, this=-
491445185, dest=queue://mq90b/JMSQueue

```

Confirmation of Queue specified in "JMSReplyTo":

This MDB program adds a hash code to the received message and forwards the message to JMSReplyTo.

Specify the connection factory and queue to the queue manager of the transfer destination and get it.

Connection Factory:

Destination:

☐ : put message

☒ : get message

The following will be displayed.

Receive message:

```
JMSMessage class: jms_text  
JMSType: null  
JMSDeliveryMode: 2  
JMSDeliveryDelay: 0  
JMSDeliveryTime: 0  
JMSExpiration: 0  
JMSPriority: 4  
JMSMessageID: ID:414d51206d7139306220202020202007c9da5923578802  
JMSTimestamp: 1507516719536  
JMSCorrelationID: null  
JMSDestination: queue://mq90b/JMSQueue  
JMSReplyTo: null  
JMSRedelivered: false  
JMSXAppID: Server\java\8.0\bin\java.exe  
JMSXDeliveryCount: 1  
JMSXUserID: okada  
JMS_IBM_Character_Set: UTF-8  
JMS_IBM_Encoding: 273  
JMS_IBM_Format: MQSTR  
JMS_IBM_MsgType: 2  
JMS_IBM_PutApplType: 11  
JMS_IBM_PutDate: 20171009  
JMS_IBM_PutTime: 02383953  
MDB test message processed by: -491445185  
SUCCESS
```

4. Receiving Non-Text Type JMS Message

Receiving a MapMessage

Set as follows and click "Submit".

JMS Message Type:

String:

Integer:

The following MapMessage will be sent from the MQRA verification program.
String name="First"(fixed) value=String specified in "String:"
String name="Second"(fixed) value=Integer specified in "Integer:"

Confirmation of SystemOut.log:

Confirm with SystemOut.log that MDB has processed the Map message.

```
[17/10/09 12:15:24:400 JST] 00000115 SystemOut   O +++ MDBFVTEJB: onMessage()
start, this=-491445185
[17/10/09 12:15:24:400 JST] 00000115 SystemOut   O +++ MDBFVTEJB:
getJMSType=CF:jms/mq90bBindCF -491445185
[17/10/09 12:15:24:400 JST] 00000115 SystemOut   O +++ MDBFVTEJB: Connection
Factory for Reply jms/mq90bBindCF -491445185
[17/10/09 12:15:24:409 JST] 00000115 SystemOut   O +++ MDBFVTEJB: JMS Message =>
```

JMSMessage class: jms_map

```
JMSType:      CF:jms/mq90bBindCF
JMSDeliveryMode: 1
JMSDeliveryDelay: 0
JMSDeliveryTime: 0
JMSExpiration: 0
JMSPriority: 0
JMSMessageID: ID:414d51206d713930202020202020202025b9da59220c0504
JMSTimestamp: 1507518924386
JMSCorrelationID: null
JMSDestination: queue:///SampleMDBQueue
JMSReplyTo:    queue://mq90b/JMSQueue
JMSRedelivered: false
JMSXAppID: Server\java\8.0\bin\java.exe
JMSXDeliveryCount: 1
JMSXUserID: okada
JMS_IBM_Character_Set: UTF-8
JMS_IBM_Encoding: 273
JMS_IBM_Format: MQSTR
JMS_IBM_MsgType: 1
JMS_IBM_PutApplType: 11
JMS_IBM_PutDate: 20171009
JMS_IBM_PutTime: 03152438
```

Second = 12345

First = map message

```
[17/10/09 12:15:24:409 JST] 00000115 SystemOut   O +++ MDBFVTEJB: getJMSReplyTo
=> queue://mq90b/JMSQueue
[17/10/09 12:15:24:409 JST] 00000115 SystemOut   O +++ MDBFVTEJB: Map Message =>
String value : map message Integer value 12345
```

```
[17/10/09 12:15:24:409 JST] 00000115 SystemOut    O +++ MDBFVTEJB: reply stream =>
String value : map message processed by: -491445185 Integer value 12345
[17/10/09 12:15:24:409 JST] 00000115 SystemOut    O +++ MDBFVTEJB: sendReply, this=-
491445185, dest=queue://mq90b/JMSQueue
```

Receiving an ObjectMessage

Set as follows and click "Submit".

JMS Message Type:

String:

Integer:

The MQRA verification program creates and sends a String object from the string specified in the String field.

Confirmation of SystemOut.log:

Confirm with SystemOut.log that MDB has processed the Object message.

```
[17/10/09 12:23:13:285 JST] 00000115 SystemOut    O +++ MDBFVTEJB: onMessage()
start, this=-491445185
[17/10/09 12:23:13:285 JST] 00000115 SystemOut    O +++ MDBFVTEJB:
getJMSType=CF:jms/mq90bBindCF -491445185
[17/10/09 12:23:13:286 JST] 00000115 SystemOut    O +++ MDBFVTEJB: Connection
Factory for Reply jms/mq90bBindCF -491445185
[17/10/09 12:23:13:294 JST] 00000115 SystemOut    O +++ MDBFVTEJB: JMS Message =>
```

JMSMessage class: jms_object

```
JMSType:      CF:jms/mq90bBindCF
JMSDeliveryMode: 1
JMSDeliveryDelay: 0
JMSDeliveryTime: 0
JMSExpiration: 0
JMSPriority:   0
JMSMessageID: ID:414d51206d713930202020202020202025b9da59220c0505
JMSTimestamp: 1507519393271
JMSCorrelationID: null
JMSDestination: queue:///SampleMDBQueue
JMSReplyTo:    queue://mq90b/JMSQueue
JMSRedelivered: false
```

JMSXAppID: Server\java\8.0\bin\java.exe
 JMSXDeliveryCount: 1
 JMSXUserID: okada
 JMS_IBM_Character_Set: UTF-8
 JMS_IBM_Encoding: 273
 JMS_IBM_Format:
 JMS_IBM_MsgType: 1
 JMS_IBM_PutApplType: 11
 JMS_IBM_PutDate: 20171009
 JMS_IBM_PutTime: 03231327

class java.lang.String

[17/10/09 12:23:13:295 JST] 00000115 SystemOut O +++ MDBFVTEJB: getJMSReplyTo
 => queue://mq90b/JMSQueue
 [17/10/09 12:23:13:295 JST] 00000115 SystemOut O +++ MDBFVTEJB: Object Message
 => object message
 [17/10/09 12:23:13:295 JST] 00000115 SystemOut O +++ MDBFVTEJB: reply object =>
 object message processed by: -491445185
 [17/10/09 12:23:13:295 JST] 00000115 SystemOut O +++ MDBFVTEJB: sendReply, this=-
 491445185, dest=queue://mq90b/JMSQueue

Receiving a StreamMessage

Set as follows and click "Submit".

JMS Message Type:

String:

Integer:

The MQRA verification program creates a StreamMessage with a string specified in "String:" as a String type, and a value specified in "Integer:" as an int type, and sends it.

Confirmation of SystemOut.log:

Confirm with SystemOut.log that MDB has processed the Stream message.

[17/10/09 12:30:51:206 JST] 00000115 SystemOut O +++ MDBFVTEJB: onMessage()
 start, this=-491445185
 [17/10/09 12:30:51:206 JST] 00000115 SystemOut O +++ MDBFVTEJB:
 getJMSType=CF:jms/mq90bBindCF -491445185
 [17/10/09 12:30:51:206 JST] 00000115 SystemOut O +++ MDBFVTEJB: Connection
 Factory for Reply jms/mq90bBindCF -491445185

[17/10/09 12:30:51:217 JST] 00000115 SystemOut O +++ MDBFVTEJB: JMS Message =>

JMSMessage class: jms_stream

JMSType: CF:jms/mq90bBindCF
JMSDeliveryMode: 1
JMSDeliveryDelay: 0
JMSDeliveryTime: 0
JMSExpiration: 0
JMSPriority: 0
JMSMessageID: ID:414d51206d713930202020202020202025b9da59220c0506
JMSTimestamp: 1507519851201
JMSCorrelationID: null
JMSDestination: queue:///SampleMDBQueue
JMSReplyTo: queue://mq90b/JMSQueue
JMSRedelivered: false
JMSXAppID: Server\java\8.0\bin\java.exe
JMSXDeliveryCount: 1
JMSXUserID: okada
JMS_IBM_Character_Set: UTF-8
JMS_IBM_Encoding: 273
JMS_IBM_Format: MQSTR
JMS_IBM_MsgType: 1
JMS_IBM_PutApplType: 11
JMS_IBM_PutDate: 20171009
JMS_IBM_PutTime: 03305120

stream message

12345

[17/10/09 12:30:51:217 JST] 00000115 SystemOut O +++ MDBFVTEJB: getJMSReplyTo
=> queue://mq90b/JMSQueue

[17/10/09 12:30:51:217 JST] 00000115 SystemOut O +++ MDBFVTEJB: Stream Message
=> String value : stream message Integer value 12345

[17/10/09 12:30:51:217 JST] 00000115 SystemOut O +++ MDBFVTEJB: reply stream =>
String value : stream message processed by: -491445185 Integer value 12345

[17/10/09 12:30:51:217 JST] 00000115 SystemOut O +++ MDBFVTEJB: sendReply, this=-
491445185, dest=queue://mq90b/JMSQueue

Receiving a ByteMessage

Set as follows and click "Submit".

JMS Message Type:

String:

Integer:

The MQRA verification program creates and transmits ByteMessage from the character string specified in "String".

Confirmation of SystemOut.log:

Confirm with SystemOut.log that MDB has processed the Byte message.

[17/10/09 12:36:32:721 JST] 00000115 SystemOut O +++ MDBFVTEJB: onMessage()
start, this=-491445185

[17/10/09 12:36:32:721 JST] 00000115 SystemOut O +++ MDBFVTEJB:
getJMSType=CF:jms/mq90bBindCF -491445185

[17/10/09 12:36:32:721 JST] 00000115 SystemOut O +++ MDBFVTEJB: Connection
Factory for Reply jms/mq90bBindCF -491445185

[17/10/09 12:36:32:730 JST] 00000115 SystemOut O +++ MDBFVTEJB: JMS Message =>

JMSMessage class: jms_bytes

JMSType: CF:jms/mq90bBindCF
 JMSDeliveryMode: 1
 JMSDeliveryDelay: 0
 JMSDeliveryTime: 0
 JMSExpiration: 0
 JMSPriority: 0
 JMSMessageID: ID:414d51206d713930202020202020202025b9da59220c0507
 JMSTimestamp: 1507520192716
 JMSCorrelationID: null
 JMSDestination: queue:///SampleMDBQueue
 JMSReplyTo: queue://mq90b/JMSQueue
 JMSRedelivered: false
 JMSXAppID: Server\java\8.0\bin\java.exe
 JMSXDeliveryCount: 1
 JMSXUserID: okada
 JMS_IBM_Character_Set: UTF-8
 JMS_IBM_Encoding: 273
 JMS_IBM_Format:
 JMS_IBM_MsgType: 1

JMS_IBM_PutApplType: 11
JMS_IBM_PutDate: 20171009
JMS_IBM_PutTime: 03363271
62797465206d657373616765

[17/10/09 12:36:32:731 JST] 00000115 SystemOut O +++ MDBFVTEJB: getJMSReplyTo
=> queue://mq90b/JMSQueue

[17/10/09 12:36:32:731 JST] 00000115 SystemOut O +++ MDBFVTEJB: Bytes Message
=> byte message

[17/10/09 12:36:32:731 JST] 00000115 SystemOut O +++ MDBFVTEJB: reply byte =>
byte message processed by: -491445185

[17/10/09 12:36:32:731 JST] 00000115 SystemOut O +++ MDBFVTEJB: sendReply, this=-
491445185, dest=queue://mq90b/JMSQueue

Conclusion

If you find any defects in this program, or if you have any questions and requests about this program, please contact us.

S.N.Software Inc. - <https://www.pulsarintegration.com>
e-mail: support@pulsarintegration.com