

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

本プログラムの 検証済み 環境

・ 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

※本プログラムは Java EE 7.0 に準拠されて以降の他の WebSphere Application Server のバージョンで実行できる可能性があります。

目次

本プログラムについて.....	4
1. プログラムの実行環境.....	5
mdbfvt のデプロイ.....	5
JNI ライブラリー情報.....	5
実行ユーザー.....	6
その他.....	6
2. テスト準備.....	7
接続ファクトリの作成.....	7
キューの作成.....	7
アクティベーション・スペックの作成.....	7
テストプログラムの起動.....	7
3. 使用方法.....	8
TextMessage の受信.....	8
4. テキスト以外の JMS Message Type のメッセージの受信.....	11
MapMessage の受信.....	11
ObectMessage の受信.....	14
StreamMessage の受信.....	16
ByteMessage の受信.....	17
おわりに	20

本プログラムについて

本プログラムは WebSphere MQ / IBM MQ およびその提供する MQ ソース・アダプターのインバウンド(MDB)の機能／使用方法を検証／確認する目的で作成されています。（アウトバウンドの機能の検証には、mqfvt (MQRA 検証プログラム)を使用してください。）下記の検証を実行することが可能です。

EJB3.1(local) コンテナ管理トランザクション、アクティベーション・スペック、JMS Message Type: TextMessage/MapMessage/ObjectMessage/StreamMessage/ByteMessage の受信。

本書では、IBM MQ 自体の詳細の解説は行いません。必要に応じて製品のマニュアルを参照してください。

全てのバージョンの製品のマニュアルが下記 URL から参照できます。

IBM MQ and WebSphere MQ

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

本プログラムテストには、MQRA 検証プログラム(mqfvt)も使用しています。それらの詳細については、「IBM MQ Java EE Connector Architecture Verification Test Program (mqfvt)」を参照してください。

1. プログラムの実行環境

mqfvt を使用する前提として、ローカル、またはリモートに IBM MQ7.0.1 以上の MQ サーバーがインストールされていること、および Java EE 7.0 に準拠したバージョンの WebSphere Application Server が使用可能になっていることが必要です。

mdbfvt のデプロイ

MDBFVTEJBEAR.ear をご利用の WebSphere Application Server にインストールすることで mqfvt は使用可能になります。

EAR のインストール前にアクティベーション・スペックが作成済みであることが必要です。作成されていない場合は、「2. テスト準備」を参考に作成します。

IBM WebSphere Application Server 9.0(Traditional) の場合の例を示します。

管理コンソールの「アプリケーション」-「新規アプリケーション」-「新規エンタープライズ・アプリケーション」で、「アプリケーション・インストールの準備」画面を開き、MDBFVTEJBEAR.ear ファイルの場所を指定します。下記以外は全てデフォルトでインストールを完了させます。

- ・「インストール・オプションの選択」で、「エンタープライズ Bean のデプロイ」をチェックする。
- ・「メッセージ駆動型 Bean のリスナーをバインド」では、「アクティベーション・スペック」をチェックし、「ターゲット・リソース JNDI 名」（アクティベーション・スペックの JNDI 名）を指定します。

JNI ライブラリー情報

キューマネージャーへの接続にバインド・モードを使用する場合、環境によっては JNI ライブラリー (mqjbnd.dll、libmqjbnd.so など) の場所を明示的に指定することが必要な場合があります。WebSphere Application Server 9.0(Traditional) では、「リソース」-「JMS」-「JMS プロバイダー」で、適切な有効範囲内にある「WebSphere MQ メッセージング・プロバイダー」を選択し、「一般プロパティ」の「ネイティブ・ライブラリー・パス」にご利用の MQ 環境に合わせて JNI ライブラリーのパスを設定します。

例)

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

実行ユーザー

アプリケーション・サーバーの実行ユーザーにキューマネージャーに設定される適切なアクセス権限が必要な場合があります。詳細は IBM MQ のドキュメントを参照してください。

その他

※一部の WAS の MQ 管理画面で、LIBPATH(AIX)やLD_LIBRARY_PATHに JNI ライブラリへのパスの設定が必要な場合があります。下記ファイル内で設定可能です。

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

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

※一部の Windows 環境で、PATH 環境変数に<MQ Install Directory>%bin64 を設定が必要な場合があります。setupCmdLine.bat(sh)で設定します。

2. テスト準備

プログラムを実行するには、接続ファクトリ、キュー、アクティベーション・スペックをアプリケーション・サーバー上に定義します。

以下は IBM WebSphere Application Server 9.0(Traditional) の場合の例です。

接続ファクトリの作成

管理コンソールの「リソース」-「JMS」-「接続ファクトリ」で、「接続ファクトリー」画面を開き、「新規作成」ボタンをクリックします。「WebSphere MQ メッセージング・プロバイダー」を選択して、「IBM MQ JMS リソースの作成」画面で、「名前」と「JNDI 名」を指定します。アプリケーションから指定する接続ファクトリ名は、ここで入力する「JNDI 名」です。（本プログラムはリソース参照を定義しない為、JNDI ENC 名は使用できません。JNDI シンプル名か JNDI 完全修飾名を指定します。）以降は適切なパラメータを指定して作成を完了します。

キューの作成

管理コンソールの「リソース」-「JMS」-「キュー」で、「キュー」画面を開き、「新規作成」ボタンをクリックします。「WebSphere MQ メッセージング・プロバイダー」を選択して、「名前」、「JNDI 名」、「キュー名」（実際のキュー名称）を指定します。（「キュー・マネージャーまたはキュー共有グループ名」を指定しない場合は、汎用的に使用できます。）アプリケーションから指定するキュー名は、ここで入力する「JNDI 名」です。以降は適切なパラメータを指定して作成を完了します。

アクティベーション・スペックの作成

管理コンソールの「リソース」-「JMS」-「アクティベーション・スペック」で、「アクティベーション・スペック」画面を開き、「新規作成」ボタンをクリックします。「WebSphere MQ メッセージング・プロバイダー」を選択して、「名前」、「JNDI 名」、「宛先 JNDI 名」（前項で定義したキューの JNDI 名）を指定します。以降は適切なパラメータを指定して作成を完了します。

テストプログラムの起動

管理コンソールの「アプリケーション」-「アプリケーション・タイプ」-「WebSphere エンタープライズ・アプリケーション」で、「エンタープライズ・アプリケーション」を開き、「MDBFVTEJB EAR」を選択（チェック）し、「開始」します。

3. 使用方法

TextMessage の受信を例に、使用方法を説明します。

TextMessage の受信

MQRA 検証プログラムより JMS メッセージを PUT します。

JMSReplyTo に"キュー URI (Uniform Resource Identifier)"、JMSType にプレフィクス"CF:"の後に接続ファクトリを指定すると受信したメッセージを転送可能です。

例：

JMSReplyTo: queue://mq90b/JMSQueue

※この値は Session.createProducer() の引数に渡されます。指定するキューの WAS 上の定義は不要です。転送されたメッセージを MQRA 検証プログラムで GET する場合は WAS 上のキュー定義が必要です。

次の様に設定し、「Submit」をクリックします。

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:

(指定内容)

IBM MQ Classes for JMS : チェック : API は JMS を使用

Connection Factory : jms/mq90BindCF : 接続ファクトリの JNDI シンプル名

Destination : jms/SampleMDBQueue : キューの JNDI 名

put message : チェック : メッセージの PUT

JMS Message Type: TextMessage : JMS のメッセージのタイプ

String : “MDB test message” : PUT するメッセージ

送信したメッセージが表示されます。

Send message:

```
JMSMessage class: jms_text
JMSType:          CF:jms/mq90bBindCF
JMSDeliveryMode:  1
JMSDeliveryDelay:  0
JMSDeliveryTime:  1507516719497
JMSExpiration:     0
JMSPriority:        0
JMSMessageID:      ID:414d51206d7139302020202020202025b9da59220c0503
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
```

SystemOut.log の確認 :

MDB がメッセージを処理したことを SystemOut.log で確認します。

```
[17/10/09 11:38:39:515 JST] 00000115 SystemOut      0 +++ MDBFVTEJB: onMessage()
start, this=-491445185

[17/10/09 11:38:39:515 JST] 00000115 SystemOut      0 +++ MDBFVTEJB:
getJMSType=CF:jms/mq90bBindCF -491445185

[17/10/09 11:38:39:516 JST] 00000115 SystemOut      0 +++ MDBFVTEJB: Connection
Factory for Reply jms/mq90bBindCF -491445185

[17/10/09 11:38:39:534 JST] 00000115 SystemOut      0 +++ MDBFVTEJB: JMS Message
```

=>

```
JMSMessage class: jms_text
JMSType:          CF:jms/mq90bBindCF
JMSDeliveryMode:  1
JMSDeliveryDelay:  0
JMSDeliveryTime:  0
JMSExpiration:    0
JMSPriority:       0
JMSMessageID:     ID:414d51206d7139302020202020202025b9da59220c0503
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      0 +++ MDBFVTEJB:
getJMSReplyTo => queue:///mq90b/JMSQueue

[17/10/09 11:38:39:535 JST] 00000115 SystemOut      0 +++ MDBFVTEJB: Text Message
=> MDB test message

[17/10/09 11:38:39:535 JST] 00000115 SystemOut      0 +++ MDBFVTEJB: reply text
=> MDB test message processed by: -491445185

[17/10/09 11:38:39:535 JST] 00000115 SystemOut      0 +++ MDBFVTEJB: sendReply,
this=-491445185, dest=queue:///mq90b/JMSQueue
```

“JMSReplyTo”に指定した Queue の確認：

本MDBプログラムは受信したメッセージに hash コードを付加し、JMSReplyTo にメッセージを転送します。

転送先のキューマネージャーへの接続ファクトリとキューを指定してGetします。

Connection Factory:

Destination:

☐ : put message

☒ : get message

下記が表示されます。

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. テキスト以外の JMS Message Type のメッセージの受信

MapMessage の受信

次の様に設定し、「Submit」をクリックします。

JMS Message Type:

String:

Integer:

MQRA 検証プログラムから下記の MapMessage が送信されます。
String name="First"(固定) value="String:"に指定した文字列
String name="Second"(固定) value="Integer:"に指定した数値

SystemOut.log の確認：

MDB が MAP メッセージを処理したことを SystemOut.log で確認します。

```
[17/10/09 12:15:24:400 JST] 00000115 SystemOut      0 +++ MDBFVTEJB:
onMessage() start, this=-491445185
[17/10/09 12:15:24:400 JST] 00000115 SystemOut      0 +++ MDBFVTEJB:
getJMSType=CF:jms/mq90bBindCF -491445185
[17/10/09 12:15:24:400 JST] 00000115 SystemOut      0 +++ MDBFVTEJB:
Connection Factory for Reply jms/mq90bBindCF -491445185
[17/10/09 12:15:24:409 JST] 00000115 SystemOut      0 +++ MDBFVTEJB: JMS
Message =>
JMSMessage class: jms_map
JMSType:          CF:jms/mq90bBindCF
JMSDeliveryMode:  1
JMSDeliveryDelay: 0
JMSDeliveryTime:  0
JMSExpiration:    0
JMSPriority:       0
JMSMessageID:     ID:414d51206d7139302020202020202025b9da59220c0504
```

```
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      0 +++ MDBFVTEJB:
getJMSReplyTo => queue://mq90b/JMSQueue
[17/10/09 12:15:24:409 JST] 00000115 SystemOut      0 +++ MDBFVTEJB: Map
Message => String value : map message Integer value 12345
[17/10/09 12:15:24:409 JST] 00000115 SystemOut      0 +++ MDBFVTEJB: reply
stream => String value : map message processed by: -491445185 Integer value
12345
[17/10/09 12:15:24:409 JST] 00000115 SystemOut      0 +++ MDBFVTEJB:
sendReply, this=-491445185, dest=queue://mq90b/JMSQueue
```

ObectMessage の受信

次の様に設定し、「Submit」をクリックします。

JMS Message Type:

String:

Integer:

MQRA 検証プログラムは、String フィールドに指定した文字列から String オブジェクトを作成して送信します。

SystemOut.logの確認：

MDBがObjectメッセージを処理したことをSystemOut.logで確認します。

```
[17/10/09 12:23:13:285 JST] 00000115 SystemOut      0 +++ MDBFVTEJB:
onMessage() start, this=-491445185
[17/10/09 12:23:13:285 JST] 00000115 SystemOut      0 +++ MDBFVTEJB:
getJMSType=CF:jms/mq90bBindCF -491445185
[17/10/09 12:23:13:286 JST] 00000115 SystemOut      0 +++ MDBFVTEJB:
Connection Factory for Reply jms/mq90bBindCF -491445185
[17/10/09 12:23:13:294 JST] 00000115 SystemOut      0 +++ 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      0 +++ MDBFVTEJB:
getJMSReplyTo => queue:///mq90b/JMSQueue
[17/10/09 12:23:13:295 JST] 00000115 SystemOut      0 +++ MDBFVTEJB: Object
Message => object message
```

```
[17/10/09 12:23:13:295 JST] 00000115 SystemOut      0 +++ MDBFVTEJB: reply
object => object message processed by: -491445185
[17/10/09 12:23:13:295 JST] 00000115 SystemOut      0 +++ MDBFVTEJB:
sendReply, this=-491445185, dest=queue://mq90b/JMSQueue
```

StreamMessage の受信

次の様に設定し、「Submit」をクリックします。

JMS Message Type:

String:

Integer:

MQRA 検証プログラムは、“String:”に指定した文字列を String 型として、“Integer:”に指定した値を int 型として StreamMessage を作成し送信します。

SystemOut.log の確認：

MDB が Stream メッセージを処理したことを SystemOut.log で確認します。

```
[17/10/09 12:30:51:206 JST] 00000115 SystemOut      0 +++ MDBFVTEJB:
onMessage() start, this=-491445185
[17/10/09 12:30:51:206 JST] 00000115 SystemOut      0 +++ MDBFVTEJB:
getJMSType=CF:jms/mq90bBindCF -491445185
[17/10/09 12:30:51:206 JST] 00000115 SystemOut      0 +++ MDBFVTEJB:
Connection Factory for Reply jms/mq90bBindCF -491445185
[17/10/09 12:30:51:217 JST] 00000115 SystemOut      0 +++ 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      0 +++ MDBFVTEJB:
getJMSReplyTo => queue://mq90b/JMSQueue
[17/10/09 12:30:51:217 JST] 00000115 SystemOut      0 +++ MDBFVTEJB: Stream
Message => String value : stream message Integer value 12345
[17/10/09 12:30:51:217 JST] 00000115 SystemOut      0 +++ MDBFVTEJB: reply
stream => String value : stream message processed by: -491445185 Integer
value 12345
[17/10/09 12:30:51:217 JST] 00000115 SystemOut      0 +++ MDBFVTEJB:
sendReply, this=-491445185, dest=queue://mq90b/JMSQueue
```

ByteMessage の受信

次の様に設定し、「Submit」をクリックします。

JMS Message Type:

String:

Integer:

MQRA 検証プログラムは、“String:”に指定した文字列から ByteMessage を作成し送信する。

SystemOut.logの確認：

MDBがByteメッセージを処理したことをSystemOut.logで確認します。

```
[17/10/09 12:36:32:721 JST] 00000115 SystemOut      0 +++ MDBFVTEJB:
onMessage() start, this=-491445185
```

```
[17/10/09 12:36:32:721 JST] 00000115 SystemOut      0 +++ MDBFVTEJB:
getMessageType=CF:jms/mq90bBindCF -491445185
```

```
[17/10/09 12:36:32:721 JST] 00000115 SystemOut      0 +++ MDBFVTEJB:
Connection Factory for Reply jms/mq90bBindCF -491445185
```

```
[17/10/09 12:36:32:730 JST] 00000115 SystemOut      0 +++ 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      0 +++ MDBFVTEJB:
getJMSReplyTo => queue://mq90b/JMSQueue

[17/10/09 12:36:32:731 JST] 00000115 SystemOut      0 +++ MDBFVTEJB: Bytes
Message => byte message

[17/10/09 12:36:32:731 JST] 00000115 SystemOut      0 +++ MDBFVTEJB: reply
byte => byte message processed by: -491445185

[17/10/09 12:36:32:731 JST] 00000115 SystemOut      0 +++ MDBFVTEJB:
sendReply, this=-491445185, dest=queue://mq90b/JMSQueue
```

おわりに

本プログラムについてのバグが見つかった場合、あるいは機能についてのご要望、その為
ご質問等がございましたら、下記までご連絡ください。

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