

# Advanced WebSphere MQ Listener Functionality

- Support for Synchronous Request/Reply Scenarios
  - Dynamic IDL/RPC Parameters for WebSphere MQ Listener
  - Support for the MQRFH Header
  - Character Encoding Issues
  - User Exit for Message Processing
  - Transactional Behavior
- 

## Support for Synchronous Request/Reply Scenarios

The WebSphere MQ Listener provides support for a synchronous request/reply scenario. In this case the remote procedure call usually has both INPUT and OUTPUT parameters, or at least one INOUT. A request/reply scenario is automatically detected by the WebSphere MQ Listener if the MQ message specifies the ReplyToQueue field. Also, if the property `entirex.wmqbridge.check.msgtype` has been set, the message type must be "MQMT\_REQUEST".

The `correlationId` of the reply is set to the `correlationId` of the request if the `report` field of the request has the flag `MQRO_PASS_CORREL_ID` set. Otherwise the `correlationId` of the reply is set to the `messageId` of the request.

The `messageId` of the reply is set to the `messageId` of the request if the `report` field of the request has the flag `MQRO_PASS_MSG_ID` set. Otherwise a new `messageId` is created.

## Dynamic IDL/RPC Parameters for WebSphere MQ Listener

With the WebSphere MQ Listener it is possible that certain parameters of a remote procedure call are dynamic parameters which are evaluated by the RPC server. Dynamic parameters have a fixed name; they can be defined only on level 1 in the parameter definition in the IDL file and before any variable length parameter, and have a specific format. The WebSphere MQ Listener uses the following parameter. All other dynamic parameters described in the corresponding section of the WebSphere MQ RPC Server documentation are ignored.

Parameter	Format	Description
MQRFH Header data (MQ_RFH_*)	A with fixed length	Arbitrary number of parameters. The names following the prefix "MQ_RFH_" are used as the names of the name value pairs of the MQRFH header.

If dynamic parameters are to be used, generate a properties file from the corresponding Software AG IDL file and specified with the `entirex.bridge.names.file` property. To generate this property file, use the template `bridge.tpl` in the `template` subdirectory of the EntireX installation. For batch generation, run `erxidl.bat` (Windows) or `erxidl.bsh` (UNIX) with the parameters "`-t <path to template directory>/bridge.tpl <idlFile>`".

Alternatively, you can also use the *EntireX Workbench*. Go to the **Preferences for EntireX** and create a new Custom Wrapper. Specify a name and browse to the `bridge.tpl` template. If the Custom Wrapper has been created (and the Workbench restarted), you can generate the properties file from an IDL file, using the context menu item **Other > Generate...**

## Support for the MQRFH Header

The WebSphere MQ Listener supports the MQRFH header (rules and formatting header). This header consists basically of name value pairs. Restriction: only one header per MQ message is possible; MQ allows an arbitrary number of headers per message.

When sending a message to MQ: an MQRFH header is built if at least one parameter in the IDL file has a name with prefix "MQ\_RFH\_". All IN (or INOUT) parameters with this prefix are used to build the header. If for example the IDL file contains two fields `MQ_RFH_H1` and `MQ_RFH_H2` with the values "v1" and "v2", the resulting MQRFH header will have two name value pairs, "H1 v1" and "H2 v2".

If a message is received from MQ: if the message has an MQRFH header, all value entries in the name value pairs are copied to the corresponding OUT (or INOUT) parameter in the IDL file. The name has to match the part of the IDL parameter name after the prefix. In the above example consider that the MQ message has two name value pairs, "H1 v11" and "H3 v22". Then the value "v11" will be assigned to the parameter `MQ_RFH_H1`, the parameter `MQ_RFH_H2` gets no value assigned, and the entry for "H3" will be ignored.

## Character Encoding Issues

If the payload of the MQ message is in XML format (property `entirex.bridge.xmm` has been set), the WebSphere MQ Listener converts the XML payload to an RPC request, using the default platform encoding of the Java virtual machine. If the WebSphere MQ Listener sends back a reply message, the XML payload of this message will be UTF-8 encoded. A different encoding can be used by setting the property `entirex.bridge.xml.encoding`.

If the payload of the MQ message is of type text (property `entirex.bridge.idl` has been set), the WebSphere MQ Listener converts the payload to an RPC request, using the default platform encoding of the Java virtual machine. If the WebSphere MQ Listener sends back a reply message, the Listener converts the payload of the message to the encoding specified by the CCSID (Coded Character Set Identification) of the MQ queue manager.

### Note:

The default platform encoding of the JVM can be changed by setting the system property `file.encoding` in the startup script of the Listener.

## User Exit for Message Processing

WebSphere MQ does not have a clearly defined message layout, it is basically a stream of bytes. In general it is up to the MQ application to know the exact semantics of an MQ message. This might include application-specific headers and formatting rules. The WebSphere MQ Listener supports a general but simplified model of message processing.

To better handle application specific message layout details, a user exit (or callback routine) can be used. The user exit works on the WebSphere MQ Java representation of an MQ message (class `com.ibm.mq.MQMessage`) and can change the MQ message. The user exit gets control:

1. after an MQ message has been read from an MQ queue and before it is processed by the WebSphere MQ Listener
2. after an MQ message has been constructed by the WebSphere MQ Listener and before the message is put to the MQ queue.

The user exit can be used, for example, for an application-specific processing of the MQRFH, MQRFH2 or even custom headers.

To enable a user exit, use the property `entirex.wmqbridge.userexit` to specify the class name of the user exit implementation. The class will be loaded using the standard classpath. You can specify a separate classpath with the property `entirex.wmqbridge.userexit.classpath`. Note that for the classpath a file or HTTP URL must be specified. Your user exit class must implement the Java interface `com.softwareag.entirex.rpcbridge.wmqBridgeExit`. This Java interface has the following methods:

```
/**
 ** This method is called after the message has been created by the
 ** WMQBridge and before the message is sent to an MQ queue (MQPUT).
 ** The Message object and/or the MessageOptions object can be changed.
 **
 ** @param msg The MQ message object.
 ** @param pmo The MQPutMessageOptions object.
 **/
public void beforePut(com.ibm.mq.MQMessage msg, com.ibm.mq.MQPutMessageOptions pmo);
/**
 ** This method is called before a message is retrieved from an MQ
 ** queue (MQGET). The MessageOptions object can be changed.
 **
 ** @param gmo The MQGetMessageOptions object.
 **/
public void beforeGet(com.ibm.mq.MQGetMessageOptions gmo);
/**
 ** This method is called after a message has been retrieved from an
 ** MQ queue (MQGET) and before the message will be processed by the
 ** WMQBridge.
 ** The Message object can be changed.
 **
 ** @param msg The MQ message object.
 **/
public void afterGet(com.ibm.mq.MQMessage msg);
```

## Transactional Behavior

The WebSphere MQ Listener always works transactionally. A message is retrieved from the queue and then passed to the RPC server application. If no error occurs, the message is committed by the WebSphere MQ Listener. If the call to the RPC server is not successful, the call is retried as specified by properties `entirex.server.retrycycles` and `entirex.bridge.retryinterval`. If all retry attempts fail, the message is rolled back and the WebSphere MQ Listener terminates. If a dead-letter queue has been specified by the property `entirex.server.deadletterqueue`, the message will be put to that queue and committed, and the Listener will not stop.