

webMethods EntireX

Listener for IBM® MQ

Version 9.12

October 2016

This document applies to webMethods EntireX Version 9.12 and all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 1997-2016 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA, Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at <http://softwareag.com/licenses>.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <http://softwareag.com/licenses/> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third-Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at <http://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

Use, reproduction, transfer, publication or disclosure is prohibited except as specifically provided for in your License Agreement with Software AG.

Document ID: EXX-MQLISTENER-912-20181116

Table of Contents

1 About this Documentation	1
Document Conventions	2
Online Information and Support	2
Data Protection	3
2 Introduction to the Listener for IBM MQ	5
Overview	6
Receiving a Message from an IBM MQ Queue	7
Sending a Message to an IBM MQ Queue	8
3 Mapping IDL Data Types to the MQ Message Buffer	9
4 Administering the Listener for IBM® MQ	11
Customizing the Listener for IBM MQ	12
Configuring the RPC Server Side	13
Configuring the IBM MQ Side	15
Starting the Listener for IBM MQ	17
Stopping the Listener for IBM MQ	17
Using SSL/TLS	17
Activating Tracing for the RPC Server	18
5 Advanced Listener for IBM MQ Functionality	19
Support for Request/Reply Scenarios	20
Character Encoding Issues	20
User Exit for Message Processing	20
Transactional Behavior	22

1

About this Documentation

■ Document Conventions	2
■ Online Information and Support	2
■ Data Protection	3

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
Monospace font	Identifies service names and locations in the format <i>folder.subfolder.service</i> , APIs, Java classes, methods, properties.
<i>Italic</i>	Identifies: Variables for which you must supply values specific to your own situation or environment. New terms the first time they occur in the text. References to other documentation sources.
Monospace font	Identifies: Text you must type in. Messages displayed by the system. Program code.
{ }	Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols.
	Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the symbol.
[]	Indicates one or more options. Type only the information inside the square brackets. Do not type the [] symbols.
...	Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...).

Online Information and Support

Software AG Documentation Website

You can find documentation on the Software AG Documentation website at <http://documentation.softwareag.com>. The site requires credentials for Software AG's Product Support site Empower. If you do not have Empower credentials, you must use the TECHcommunity website.

Software AG Empower Product Support Website

If you do not yet have an account for Empower, send an email to empower@softwareag.com with your name, company, and company email address and request an account.

Once you have an account, you can open Support Incidents online via the eService section of Empower at <https://empower.softwareag.com/>.

You can find product information on the Software AG Empower Product Support website at <https://empower.softwareag.com>.

To submit feature/enhancement requests, get information about product availability, and download products, go to [Products](#).

To get information about fixes and to read early warnings, technical papers, and knowledge base articles, go to the [Knowledge Center](#).

If you have any questions, you can find a local or toll-free number for your country in our Global Support Contact Directory at https://empower.softwareag.com/public_directory.asp and give us a call.

Software AG TECHcommunity

You can find documentation and other technical information on the Software AG TECHcommunity website at <http://techcommunity.softwareag.com>. You can:

- Access product documentation, if you have TECHcommunity credentials. If you do not, you will need to register and specify "Documentation" as an area of interest.
- Access articles, code samples, demos, and tutorials.
- Use the online discussion forums, moderated by Software AG professionals, to ask questions, discuss best practices, and learn how other customers are using Software AG technology.
- Link to external websites that discuss open standards and web technology.

Data Protection

Software AG products provide functionality with respect to processing of personal data according to the EU General Data Protection Regulation (GDPR). Where applicable, appropriate steps are documented in the respective administration documentation.

2 Introduction to the Listener for IBM MQ

■ Overview	6
■ Receiving a Message from an IBM MQ Queue	7
■ Sending a Message to an IBM MQ Queue	8

The Listener for IBM® MQ runs as a listener on an IBM MQ queue and passes messages to an RPC server. It is used to send messages received from an IBM MQ queue to an RPC server application. This means that existing RPC servers can be used for communication with IBM MQ. This chapter covers the following topics:

Overview

The Listener for IBM® MQ receives asynchronous and synchronous messages from an IBM MQ queue and calls a standard RPC server. The Listener for IBM MQ uses the IBM® MQ base Java classes from IBM. It can connect to an IBM MQ either as an IBM MQ client using TCP/IP (*client mode*) or in so-called *bindings mode* where it is connected directly to IBM® MQ running on the same machine. Note that on z/OS, only bindings mode is supported. If the Listener for IBM MQ wants to connect in client mode via TCP/IP to an MQ server on z/OS, the *client attachment feature* needs to be installed on the target queue manager.

The Listener for IBM MQ runs as a listener on an MQ queue and processes MQ messages. It receives an MQ message and sends the message to an RPC server. A synchronous scenario is possible if the MQ message is a request message that specifies a reply queue. In this case the result returned by the RPC server is sent back as an MQ message to the reply queue.

The Listener for IBM MQ can process MQ messages in the following formats:

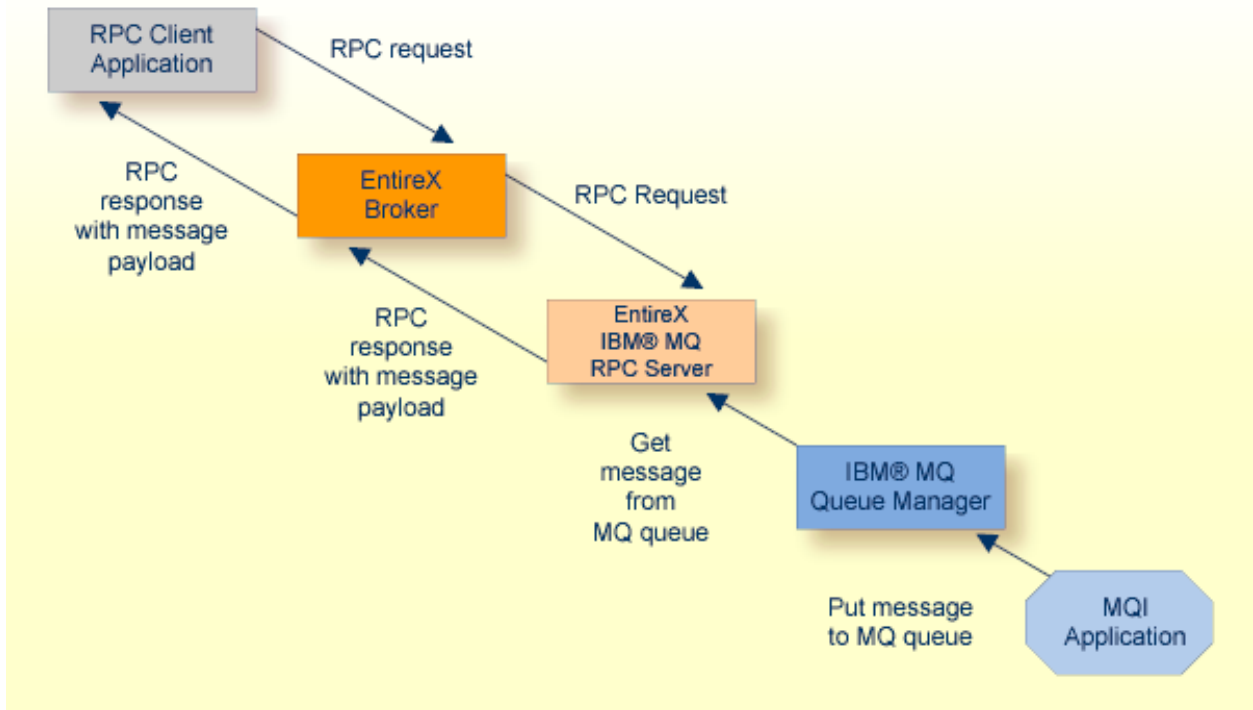
■ XML/SOAP

MQ messages in XML or SOAP format can be mapped to a Software AG IDL file, using XML Mapping Editor. The resulting XMM file has to be specified by the configuration property `entirex.bridge.xmm`. EntireX configuration parameters inside the XML/SOAP payload are ignored. See *XML/SOAP Listener* under *Writing Advanced Applications with the XML/SOAP Wrapper*.

■ Text

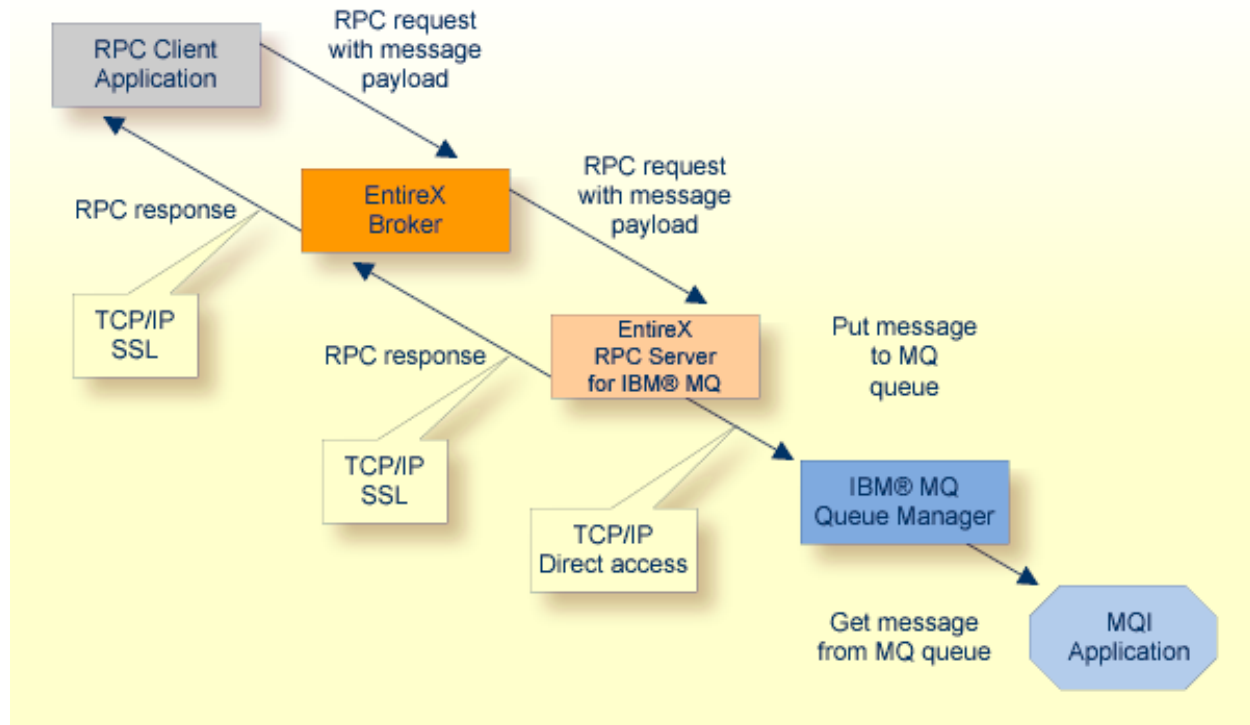
MQ messages in text format use an IDL file, which describes the message layout (see [Mapping IDL Data Types to the MQ Message Buffer](#)). The IDL file has to be specified by the configuration property `entirex.bridge.idl`. The program name is taken from the `applicationIdData` field of the incoming MQ message. If this field is empty and the IDL file has only one program this program will be called. A custom logic (e.g. using the first *n* bytes of the MQ message payload) can be implemented in the user exit.

Receiving a Message from an IBM MQ Queue



Note: All messages retrieved by the Listener for IBM MQ from the MQ input queue are passed to the same RPC service. Messages are retrieved in the order they appear on the queue.

Sending a Message to an IBM MQ Queue



Note: All messages sent to a Listener for IBM MQ instance via a specific RPC service are put on the same MQ output queue.

3 Mapping IDL Data Types to the MQ Message Buffer

The Listener for IBM MQ uses a predefined mapping of IDL data types to the MQ message buffer if the MQ message is in text format.

If your programs use arrays of groups, you have to set the property `entirex.bridge.marshalling` to "Natural" or "COBOL". If your programs do not use arrays of groups, you must not set `entirex.bridge.marshalling`.

Data Type	Description	Format	Note
<i>A</i> <i>number</i>	Alphanumeric	<i>number</i> bytes, encoding the characters.	
AV	Alphanumeric variable length	Bytes up to the end of the buffer.	1, 4
AV[<i>number</i>]	Alphanumeric variable length with maximum length	Bytes up to the end of the buffer, maximal length <i>number</i> .	1
<i>K</i> <i>number</i>	Kanji	Same as data type A.	
KV	Kanji variable length	Same as data type AV.	1, 4
KV[<i>number</i>]	Kanji variable length with maximum length	Same as data type AV[<i>number</i>].	1
I1	Integer (small)	<i>sign</i> (+, -) and 3 bytes (digits).	
I2	Integer (medium)	<i>sign</i> (+, -) and 5 bytes (digits).	
I4	Integer (large)	<i>sign</i> (+, -) and 10 bytes (digits).	
N <i>number1</i> [. <i>number2</i>]	Unpacked decimal	<i>sign</i> (+, -), <i>number1</i> bytes (digits) [<i>number2</i>] bytes (digits), no decimal point.	
P <i>number1</i> [. <i>number2</i>]	Packed decimal	<i>sign</i> (+, -), <i>number1</i> bytes (digits) [<i>number2</i>] bytes (digits), no decimal point.	
L	Logical	1 byte: X for true, all other false.	
D	Date	YYYYMMDD.	2
T	Time	YYYYMMDDhhmmssS.	3



Notes:

1. Only as last value.
2. *YYYY* year, *MM* month, *DD* day.
3. *YYYY* year, *MM* month, *DD* day, *hh* hour, *mm* minute, *ss* second, *S* tenth of a second.
4. Not possible when using COBOL.

Data types not supported:

- Binary ($B[n]$, BV , $BV[n]$)
- Floating point ($F4$, $F8$)

4

Administering the Listener for IBM® MQ

■ Customizing the Listener for IBM MQ	12
■ Configuring the RPC Server Side	13
■ Configuring the IBM MQ Side	15
■ Starting the Listener for IBM MQ	17
■ Stopping the Listener for IBM MQ	17
■ Using SSL/TLS	17
■ Activating Tracing for the RPC Server	18

The Listener for IBM® MQ runs as a listener on an IBM MQ queue and passes messages to an RPC server. It is used to send messages received from an IBM MQ queue to an RPC server application. This means that existing RPC servers can be used for communication with IBM MQ.

Customizing the Listener for IBM MQ

To set up the Listener for IBM MQ, there is a configuration file and there are scripts to start the Listener for IBM MQ.

The Listener for IBM MQ is contained in *entirex.jar*. There are two parts: the RPC side and the IBM® MQ side.

The Listener for IBM MQ uses the IBM® MQ base Java classes from IBM. To run the Listener for IBM MQ, you need either the base Java classes or a full installation of IBM® MQ. Prerequisites for all EntireX components are described centrally. See *Prerequisites for RPC Server and Listener for IBM® MQ* in the respective section of the EntireX Release Notes for the required JAR file(s). The IBM® MQ environment variables `MQ_JAVA_LIB_PATH` and `MQ_JAVA_INSTALL_PATH` must be set.

Make sure that either the local IBM® MQ installation or the IBM® MQ Java classes are accessible.

The default name for the configuration file is *entirex.wmqbridgelistener.properties*. The Listener for IBM MQ searches for this file in the current working directory. You can set the name of the configuration file with `-Dentirex.server.properties= your file name`. Use the slash “/” as file separator. The configuration file contains the configuration for both parts of the Listener for IBM MQ.

Alternatively, use script `wmqbridgelistener.bsh` (UNIX) or `wmqbridgelistener.bin` in the *bin* directory to start the Listener for IBM MQ. Both scripts use the configuration file *entirex.wmqbridgelistener.properties* in the *etc* directory, and both can be customized.

Configuring the RPC Server Side

The Listener for IBM MQ converts an MQ message into an RPC request to an RPC server. The RPC server is defined using the following properties.

Name	Default Value	Explanation								
entirex.bridge.marshalling		Define the type of marshalling (Natural or COBOL). Must be set only if the IDL file contains arrays of groups. See Mapping IDL Data Types to the MQ Message Buffer .								
entirex.server.brokerid	localhost	Broker ID. See <i>URL-style Broker ID</i> .								
entirex.server.serveraddress	RPC/SRV1/CALLNAT	Server address								
entirex.server.userid	mqListener	The Broker user ID.								
entirex.server.libname		The RPC library name (optional). The default value is the library name of the XMM/IDL file.								
entirex.server.naturallogon	no	Enables or disables logon to Natural Security for Natural RPC Server.								
entirex.server.reliableRPC	no	If set to "yes", use reliable RPC for the call to the RPC Server.								
entirex.server.rpcuser		Optional. RPC user ID (if different from entirex.server.brokerid).								
entirex.server.rpcpassword		Optional. RPC password (if different from entirex.server.password).								
entirex.server.retrycycles	15	Number of retry attempts if the call to the RPC server is not successful. If all attempts fail, the MQ message will not be committed and the Listener for IBM MQ will terminate. If a dead-letter queue has been specified, the message will be put to that queue and committed and the Listener will not stop.								
entirex.server.retryinterval	20	Retry interval (in seconds) if the call to the RPC server is not successful.								
entirex.server.compresslevel	0 (no compression)	<table><tr><td>Permitted values (you can enter the text or the numeric value):</td><td></td></tr><tr><td>BEST_COMPRESSION</td><td>9</td></tr><tr><td>BEST_SPEED</td><td>1</td></tr><tr><td>DEFAULT_COMPRESSION</td><td>-1, mapped to 6</td></tr></table>	Permitted values (you can enter the text or the numeric value):		BEST_COMPRESSION	9	BEST_SPEED	1	DEFAULT_COMPRESSION	-1, mapped to 6
Permitted values (you can enter the text or the numeric value):										
BEST_COMPRESSION	9									
BEST_SPEED	1									
DEFAULT_COMPRESSION	-1, mapped to 6									

Name	Default Value	Explanation
		DEFLATED 8
		NO_COMPRESSION 0
		N 0
		Y 8
	-help	Display usage of the command-line parameters.
entirex.server.logfile		Name of the log file, the default is standard output.
entirex.server.password		The password for secured access to the Broker. The password is encrypted and written to the property <code>entirex.server.password.e</code> . To change the password, set the new password in the properties file (default is <code>entirex.wmqbridge.properties</code>). To disable password encryption, set <code>entirex.server.passwordencrypt=no</code> (default for this property is yes).
entirex.server.properties	entirex.wmqbridgelistener.properties	The file name of the property file.
entirex.server.security	no	no/yes/auto/Name of BrokerSecurity object.
entirex.server.logfile		Name of the log file, the default is standard output.
entirex.server.verbose	no	Enable verbose output to the log file.
entirex.server.waitserver	60S	Wait time for the call to the RPC server.
entirex.timeout	20	TCP/IP transport timeout. See <i>Setting the Transport Timeout</i> under <i>Writing Advanced Applications - EntireX Java ACI</i> .
entirex.trace	0	Trace level (1,2,3).

Configuring the IBM MQ Side

These properties are used to configure the connection to the IBM® MQ queue manager.

Name	Default Value	Explanation
entirex.wmqbridge.host		If host is not specified, bindings mode is used to connect to the local MQ Server. Otherwise specify the hostname or IP address of the MQ Server.
entirex.wmqbridge.port	1414	Port of the MQ Server. Not used in bindings mode.
entirex.wmqbridge.channel	SYSTEM.DEF.SVRCONN	Channel name used to the MQ Server. Not used in bindings mode.
entirex.wmqbridge.queuemanager		Name of the (local or remote) queue manager. If not specified, a connection is made to the default queue manager.
entirex.wmqbridge.listenqueue		Name of the queue from which messages are retrieved.
entirex.wmqbridge.userid		User ID for MQ Server.
entirex.wmqbridge.password		Password for MQ Server.
entirex.wmqbridge.userexit		Class name for WMQBridge user exit.
entirex.wmqbridge.userexit.classpath		URL of the classpath for WMQBridge user exit (optional).
entirex.wmqbridge.ccsid	platform encoding	Coded Character Set Identification used by the Listener for IBM MQ (which acts as an MQ client), unused in bindings mode.

Name	Default Value	Explanation
entirex.wmqbridge.mqtrace	0	MQ tracing enabled if parameter > 0.
entirex.bridge.idl		Name of a Software AG IDL file; messages to/from MQ are in plain text.
entirex.bridge.xmm		Name of XMM (XML mapping) file; messages to/from MQ will be converted to XML.
entirex.bridge.xml.encoding		Encoding of the reply XML document.
entirex.bridge.verbose	no	Verbose/trace mode of Listener for IBM MQ
entirex.wmqbridge.environment.sslCipherSuite		Configuration for SSL connection to MQ Server. See the IBM® MQ documentation for details.
entirex.wmqbridge.environment.sslFipsRequired		Configuration for SSL connection to MQ Server. See the IBM® MQ documentation for details.
entirex.wmqbridge.priority		Message priority for messages sent to MQ (different from the default priority of the destination queue).
entirex.wmqbridge.deadletterqueue		Name of the queue that will receive unprocessed messages.

Starting the Listener for IBM MQ

Use start script `mqListener.bsh` (UNIX) or `mqListener.bat` (Windows) in the *bin* directory to start the Listener for IBM MQ. You may customize this file. See also *Prerequisites for RPC Server and Listener for IBM® MQ* in the respective section of the EntireX Release Notes.

The start scripts contain references to JAR files in the WS-Stack directory. If you update these JAR files, you may need to customize the JAR file names in the script files.

Stopping the Listener for IBM MQ

Use CTRL-C to stop the Listener for IBM MQ.

Using SSL/TLS

To use SSL with the Listener for IBM MQ, you need to configure two sides:

- **IBM® MQ Side**

See parameters `entirex.wmqbridge.environment.sslCipherSuite` and `entirex.wmqbridge.environment.sslFipsRequired` under [Configuring the IBM MQ Side](#).

- **RPC Server side**

RPC servers can use Secure Sockets Layer/Transport Layer Security (SSL/TLS) as the transport medium. The term “SSL” in this section refers to both SSL and TLS. RPC-based servers are always SSL clients. The SSL server can be either the EntireX Broker, Broker SSL Agent, or Direct RPC in webMethods Integration Server (IS inbound). For an introduction see *SSL/TLS and Certificates with EntireX* in the EntireX Security documentation.

➤ **To use SSL**

- 1 To operate with SSL, certificates need to be provided and maintained. Depending on the platform, Software AG provides default certificates, but we strongly recommend that you create your own. See *SSL/TLS Sample Certificates Delivered with EntireX* in the EntireX Security documentation.
- 2 Set up the Listener for IBM MQ for an SSL connection.

Use the *URL-style Broker ID* with protocol `ssl://` for the Broker ID. If no port number is specified, port 1958 is used as default. Example:

```
ssl://localhost:22101?trust_store=C:\SoftwareAG\EntireX\etc\ExxCACert.jks?verify_server=no
```

If the SSL client checks the validity of the SSL server only, this is known as *one-way SSL*. The mandatory `trust_store` parameter specifies the file name of a keystore that must contain the list of trusted certificate authorities for the certificate of the SSL server. By default a check is made that the certificate of the SSL server is issued for the hostname specified in the Broker ID. The common name of the subject entry in the server's certificate is checked against the hostname. If they do not match, the connection will be refused. You can disable this check with SSL parameter `verify_server=no`.

If the SSL server additionally checks the identity of the SSL client, this is known as *two-way SSL*. In this case the SSL server requests a client certificate (the parameter `verify_client=yes` is defined in the configuration of the SSL server). Two additional SSL parameters must be specified on the SSL client side: `key_store` and `key_passwd`. This keystore must contain the private key of the SSL client. The password that protects the private key is specified with `key_passwd`.

The ampersand (&) character cannot appear in the password.

SSL parameters are separated by ampersand (&). See also *SSL/TLS Parameters for SSL Clients*.

- 3 Make sure the SSL server to which the RPC side connects is prepared for SSL connections as well. The SSL server can be EntireX Broker, Broker SSL Agent, or Direct RPC in webMethods Integration Server (IS inbound). See:
 - *Running Broker with SSL/TLS Transport* in the platform-specific Administration documentation
 - *Setting up and Administering the EntireX Broker SSL Agent* in the UNIX and Windows Administration documentation
 - *Support for SSL/TLS* in the EntireX Adapter documentation (for Direct RPC)

Activating Tracing for the RPC Server

The trace level for the EntireX RPC part is controlled by the usual `entirex.trace` property. It can be set in the configuration file. Additional diagnostic output can be enabled by setting the property `entirex.server.verbose`.

The Listener for IBM MQ-specific diagnostic output is enabled by setting the property `entirex.bridge.verbose`. In addition, tracing of the IBM® MQ classes can be influenced with the property `entirex.wmqbridge.mqtrace`.

Redirect the trace to a file with the property `entirex.server.logfile`. Set this to the file name of the log file.

5

Advanced Listener for IBM MQ Functionality

■ Support for Request/Reply Scenarios	20
■ Character Encoding Issues	20
■ User Exit for Message Processing	20
■ Transactional Behavior	22

Support for Request/Reply Scenarios

The Listener for IBM MQ 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 Listener for IBM MQ 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.

Character Encoding Issues

If the payload of the MQ message is in XML format (property `entirex.bridge.xmm` has been set), the Listener for IBM MQ converts the XML payload to an RPC request, using the default platform encoding of the Java virtual machine. If the Listener for IBM MQ 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 Listener for IBM MQ converts the payload to an RPC request, using the default platform encoding of the Java virtual machine. If the Listener for IBM MQ 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

IBM® 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 Listener for IBM MQ 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 IBM® 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 Listener for IBM MQ
2. after an MQ message has been constructed by the Listener for IBM MQ 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 Listener for IBM MQ 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 Listener for IBM MQ. 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 Listener for IBM MQ 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.