

webMethods EntireX

IMS Connect RPC Server

Innovation Release

Version 9.9

October 2015

This document applies to webMethods EntireX Version 9.9 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 © 1997-2015 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.

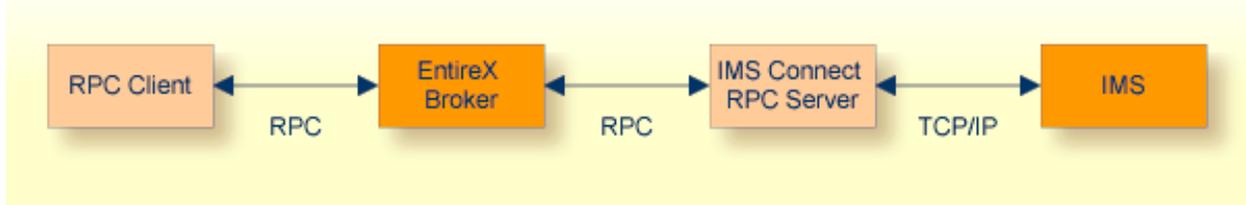
Document ID: EXX-IMSCONNECT-99-20171128

Table of Contents

1 Introduction to EntireX IMS Connect RPC Server	1
2 Administrating EntireX IMS Connect RPC Server	3
Customizing the IMS Connect RPC Server	4
Configuring the RPC Server Side	6
Configuring the IMS Connect Side	9
Using SSL/TLS with the RPC Server	10
Starting the IMS Connect RPC Server	12
Stopping the IMS Connect RPC Server	12
Application Identification	13
3 Extracting from Message Format Service	15
4 Server-side Mapping Files	17
Server-side Mapping Files in the RPC Server	18
Deploying Server-side Mapping Files to the RPC Server	18
Undeploying Server-side Mapping Files from the RPC Server	19
Change Management of Server-side Mapping Files	19
List Deployed Server-side Mapping Files	19
Check if a Server-side Mapping File Revision has been Deployed	19
Is There a Way to Smoothly Introduce Server-side Mapping Files?	20
5 Scenarios	21
COBOL Scenarios	22

1 Introduction to EntireX IMS Connect RPC Server

The EntireX IMS Connect RPC Server allows standard RPC clients to communicate with IMS MPP programs running on IMS version 9 and higher. The IMS Connect RPC Server transforms the RPCs from the clients into messages to IMS Connect v2.2. The IMS Connect RPC Server acts on one side as an RPC server and on the other side as a client for IMS Connect. The IMS Connect RPC Server is a Java-based component that can run on a different host to the one where IMS is running. This allows it to operate with a zero footprint of EntireX on the IMS host.



For existing COBOL MPP programs you can use the *Software AG IDL Extractor for COBOL* to generate the IDL file for the RPC clients. The list under *DATA DIVISION Mapping* under *COBOL to IDL Mapping* in the IDL Extractor for COBOL documentation discusses which clauses of COBOL syntax are supported and how these are handled by the extractor.

All source COBOL files have to exist locally for the IDL Extractor for COBOL. Remote extraction is not possible if there is no EntireX RPC server (batch or IMS) with extractor service enabled on the IMS host.

The IMS Connect RPC Server supports RPC clients in different programming languages.

2 Administrating EntireX IMS Connect RPC Server

▪ Customizing the IMS Connect RPC Server	4
▪ Configuring the RPC Server Side	6
▪ Configuring the IMS Connect Side	9
▪ Using SSL/TLS with the RPC Server	10
▪ Starting the IMS Connect RPC Server	12
▪ Stopping the IMS Connect RPC Server	12
▪ Application Identification	13

The EntireX IMS Connect RPC Server allows standard RPC clients to communicate with IMS MPP programs running on IMS version 9 and higher. The IMS Connect RPC Server transforms the RPCs from the clients into messages to IMS Connect v2.2. The IMS Connect RPC Server acts on one side as an RPC server and on the other side as a client for IMS Connect. The IMS Connect RPC Server is a Java-based component that can run on a different host to the one where IMS is running. This allows it to operate with a zero footprint of EntireX on the IMS host.

Customizing the IMS Connect RPC Server

The following are provided to set up the IMS Connect RPC Server:

- a configuration file
- scripts to start the IMS Connect RPC Server.

Location of the IMS Connect RPC Server

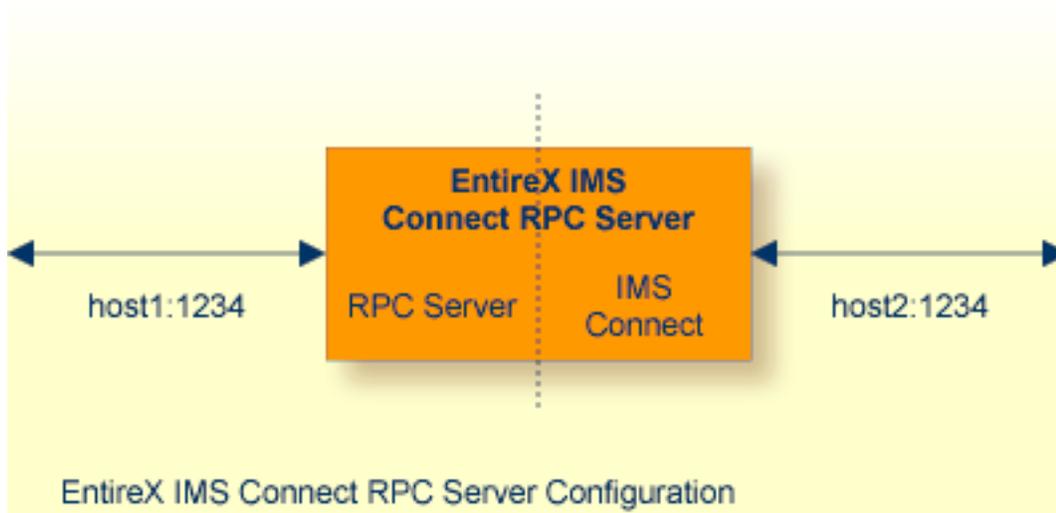
The IMS Connect RPC Server is contained in the file *entirex.jar*.

The Configuration File

The default name of the configuration file is *entirex.imsconnect.properties*. The IMS Connect RPC Server 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>` with `/` as file separator.

The configuration file contains the configuration for both parts of the IMS Connect RPC Server.



➤ To set up the IMS Connect RPC Server

- 1 Use the RPC server agent of the System Management Hub.
- 2 Add the IMS Connect RPC Server as an RPC server.

See *Administering the EntireX RPC Servers using System Management Hub* in the UNIX and Windows administration documentation for details.

Or:

Use the scripts to start the IMS Connect RPC Server.

Under Windows, use *imsconnectserver.bat* in the folder *bin* to start the IMS Connect RPC Server. You may customize this file.

Under UNIX, use *imsconnectserver.bsh* in the folder *bin* to start the IMS Connect RPC Server. You may customize this file.

Both scripts use the configuration file *entirex.imsconnect.properties* in the folder *etc*.

Configuring more than one IMS Connect RPC Server

If you configure more than one IMS Connect RPC Server that connects to the same broker, the following items must be distinct:

- the trace output file (property `entirex.server.logfile`)
- the monitor port for SMH (property `entirex.server.monitorport`)
- the log for the Windows Service (property `entirex.server.serverlog`)
- the trace output file of the SMH agent for RPC servers

Configuring the RPC Server Side

The RPC server side of the IMS Connect RPC Server is configured like the Java RPC Server. The IMS Connect RPC Server uses the properties that start with “entirex.server”.

The RPC server side can adjust the number of worker threads to the number of parallel requests. Use the properties `entirex.server.fixedservers`, `entirex.server.maxservers` and `entirex.server.minservers` to configure this scalability.

- If `entirex.server.fixedservers=yes`, the number of `entirex.server.minservers` is started and the server can process this number of parallel requests.
- If `entirex.server.fixedservers=no`, the number of worker threads balances between `entirex.server.minservers` and `entirex.server.maxservers`. This is done by a so-called attach server thread. On startup, the number of worker threads is `entirex.server.minservers`.

If more than `entirex.server.minservers` are waiting for requests, a worker thread stops if its receive call times out. The timeout period is configured with `entirex.server.waitserver`.

Alternatively to the properties, you can use command-line options. These have a higher priority than the properties set as Java system properties, and these have higher priority than the properties in the configuration file.

Name	Command-line Option	Default Value	Explanation
<code>entirex.bridge.verbose</code>		no	Verbose/trace mode of IMS Connect RPC Server
<code>entirex.server.brokerid</code>	<code>-broker</code>	localhost	Broker ID. See <i>URL-style Broker ID</i> .
<code>entirex.server.serveraddress</code>	<code>-server</code>	RPC/SRV1/CALLNAT	Server address.
<code>entirex.server.userid</code>	<code>-user</code>	IMSRPCServer	The user ID for the Broker for RPC. See <code>entirex.server.password</code> .
<code>entirex.server.fixedservers</code>		no	no Use attach server to manage worker threads. yes Run minimum number of server threads.
<code>entirex.server.minservers</code>		1	Minimum number of server threads.
<code>entirex.server.maxservers</code>		32	Maximum number of server threads.
<code>entirex.server.restartcycles</code>	<code>-restartcycles</code>	15	Number of restart attempts if the Broker is not available. This can be used to keep the IMS Connect RPC Server running

Name	Command-line Option	Default Value	Explanation														
			while the Broker is down for a short time.														
entirex.server.password	-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.imsconnect.properties</code>). To disable password encryption set <code>entirex.server.passwordencrypt=no</code> . Default for this property is "yes".														
entirex.server.properties	-propertyfile	entirex.server.properties	The file name of the property file.														
entirex.server.security	-security	no	<code>nolyslautolname of BrokerSecurity object</code>														
entirex.server.encryptionlevel		0	Deprecated. For encrypted transport we strongly recommend using the Secure Sockets Layer/Transport Layer Security protocol. See <i>SSL/TLS and Certificates with EntireX</i> .														
entirex.server.compresslevel	-compresslevel	0	Permitted values (you can enter the text or the numeric value) <table border="0"> <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> <tr><td>DEFLATED</td><td>8</td></tr> <tr><td>NO_COMPRESSION</td><td>0</td></tr> <tr><td>N</td><td>0</td></tr> <tr><td>Y</td><td>8</td></tr> </table>	BEST_COMPRESSION	9	BEST_SPEED	1	DEFAULT_COMPRESSION	-1, mapped to 6	DEFLATED	8	NO_COMPRESSION	0	N	0	Y	8
BEST_COMPRESSION	9																
BEST_SPEED	1																
DEFAULT_COMPRESSION	-1, mapped to 6																
DEFLATED	8																
NO_COMPRESSION	0																
N	0																
Y	8																
entirex.server.waitattach		600S	Wait timeout for the attach server thread.														
entirex.server.waitserver		300S	Wait timeout for the worker threads.														
entirex.timeout		20	TCP/IP transport timeout. See <i>Setting the Transport Timeout</i> under <i>Writing Advanced Applications - EntireX Java ACI</i> .														
entirex.server.verbose	-verbose	no	Enable verbose output to the log file.														

Name	Command-line Option	Default Value	Explanation
entirex.server.logfile	-logfile		Name of the log file, the default is standard output.
entirex.trace	-trace	0	Trace level (1,2,3).
entirex.server.monitorport	-smhport	0	The port where the server listens for commands from the System Management Hub (SMH). If this port is 0, no port is used and management by the SMH is disabled.

Configuring the IMS Connect Side

These properties are used to configure the connection to IMS Connect.

Alternatively, you can use the command-line option. The command-line options have a higher priority than the properties set as Java system properties and these have higher priority than the properties in the configuration file

Name	Default Value	Explanation
ims.host		Host name of IMS Connect. Mandatory.
ims.port		Port number of IMS Connect. Mandatory.
ims.datastoreid		Data store ID. Name of the IMS system that will receive transactions. Mandatory.
entirex.bridge.targetencoding	cp037	Specify the appropriate EBCDIC encoding used by your IMS Connect. This codepage is also used when communicating with the EntireX Broker. Note: Enable conversion in the Broker attribute file so the data can be converted correctly, typically by setting service-specific attribute CONVERSION to "SAGTCHA". Default "cp037" is EBCDIC codepage with full Latin-1 character set.
ims.useoldexit	yes	yes Use old IMS Connect user message exit. Name is *SAMPLE*. no Use new IMS Connect user message exit. Name is *SAMPLE1*.
ims.exitname	*SAMPLE* (old exit) *SAMPL1* (new exit)	Name of IMS Connect user message exit.
ims.sockettimeout	10000	Socket timeout for connection to IMS Connect (in milliseconds).
ims.checkdfs	true	true, yes Check for DFS message. Return an error and do not return the message if it contains a DFS error message. false, no Do not check for DFS message.
ims.clientid		ID of the client that is used by IMS Connect. Maximum 8 bytes (optional).

Name	Default Value	Explanation
ims.lterm		IMS LTERM override. Maximum 8 bytes (optional).
ims.userid		RACF user ID. Maximum 8 bytes (optional).
ims.groupid		RACF group ID. Maximum 8 bytes (optional).
ims.password		RACF password/PassTicket. Maximum 8 bytes (optional).
ims.applname		RACF application name. Maximum 8 bytes (optional).
ims.sslparams		SSL parameters (optional). Same syntax as Broker ID.
ims.mapping.folder		<p>The folder where the RPC server expects server-side mapping files (EntireX Workbench files with extension .svm). See Deploying Server-side Mapping Files to the RPC Server and Undeploying Server-side Mapping Files from the RPC Server.</p> <p>There are also client-side mapping files that do not require configuration here. See <i>Server Mapping Files for COBOL</i>.</p> <p>If <i>no</i> server requires server-side mapping, you can omit this property.</p> <p>If <i>one</i> server requires server-side mapping, this property must be specified.</p>
ims.useprogramname	false	Automatically use the IDL program name as transaction name. If set to "true" or "yes", 10 bytes are used for the transaction name. If set to a number, this number of bytes is used for the transaction name.

Using SSL/TLS with the RPC Server

To use SSL with the IMS Connect RPC Server, you need to configure two sides:

- **IMS Connect Side**

See parameter `ims.sslparams` under [Configuring the IMS Connect 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 or Broker SSL Agent. For an introduction see SSL/TLS and Certificates in the 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 *Default Certificates Delivered with EntireX*.
- 2 Set up the IMS Connect RPC Server 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 EntireX Clients and Servers*.

- 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 or Broker SSL Agent. 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

Starting the IMS Connect RPC Server

➤ To start the IMS Connect RPC Server

- Use the script `imsconnectserver` in the folder `bin` to start the IMS Connect RPC Server. You may customize this file.

Or:

Use the RPC server agent in the System Management Hub to configure and start the IMS Connect RPC Server.

See *Administering the EntireX RPC Servers using System Management Hub* in the UNIX and Windows administration documentation for details.

Stopping the IMS Connect RPC Server

➤ To stop the IMS Connect RPC Server

- Use the RPC server agent in the SMH to stop the IMS Connect RPC Server.

Or:

Use the agent for the Broker. Use `Deregister` on the service, specified with the property `entirex.server.serveraddress`.

Application Identification

The application identification is sent from the IMS Connect RPC Server to the Broker. It is visible with Broker Command and Info Services.

The identification consists of four parts: name, node, type, and version. These four parts are sent with each Broker call and are visible in the trace information.

For the IMS Connect RPC Server, these values are:

Identification Part	Value
Application name	ANAME=IMS Connect RPC Server
Node name	ANODE=<host name>
Application type	ATYPE=Java
Version	AVERS=9.9.0.0

3

Extracting from Message Format Service

To extract interface definitions from Message Format Service (MFS) with MID and MOD definitions, use a command line extractor. This extractor is run with the following command line:

```
java -classpath <suite installation ↵  
folder>\IntegrationServer\packages\WmEntireX\code\jars\entirex.jar ↵  
com.softwareag.entirex.ims.extractor.MFSExtractor <inputfile>
```

The input file has to be an MFS source with MID and MOD definitions. The output is an IDL file with the same name as the input file and suffix ".idl". Use the IDL file to generate connections and adapter services with the *Integration Server Wrapper*.

4 Server-side Mapping Files

- Server-side Mapping Files in the RPC Server 18
- Deploying Server-side Mapping Files to the RPC Server 18
- Undeploying Server-side Mapping Files from the RPC Server 19
- Change Management of Server-side Mapping Files 19
- List Deployed Server-side Mapping Files 19
- Check if a Server-side Mapping File Revision has been Deployed 19
- Is There a Way to Smoothly Introduce Server-side Mapping Files? 20

Server mapping enables the RPC server to correctly support special COBOL syntax such as `REDEFINES`, `SIGN LEADING` and `OCCURS DEPENDING ON` clauses, `LEVEL-88` fields, etc. If one of these elements is used, the IDL Extractor for COBOL automatically extracts a server mapping file in addition to the IDL file (interface definition language). Also, the COBOL Wrapper may generate a server mapping file for RPC server generation. The server mapping is used at runtime to marshal and unmarshal the RPC data stream. There are client-side mapping files (EntireX Workbench files with extension `.cvm`) and server-side mapping files (Workbench files with extension `.svm`). If you have not used server-side mapping, we recommend you use client-side mapping. See *Server Mapping Files for COBOL* in the EntireX Workbench documentation.

See also *Source Control of Server Mapping Files* | *Comparing Server Mapping Files* | *When is a Server Mapping File Required?* | *Migrating Server Mapping Files* in the EntireX Workbench documentation.

Server-side Mapping Files in the RPC Server

For an IMS Connect RPC Server, server mapping information is contained in a server-side mapping file (EntireX Workbench file with extension `.svm`) See *Server Mapping Files for COBOL*. Server mapping files are provided as operating system files in an RPC server related server-side mapping container (directory or folder). The files have the same format as in the Workbench. See *Configuring the IMS Connect Side*.

If *no* server requires a server mapping file, you can omit the property `ims.mapping.folder`.

If *one* server requires a server mapping file, provide the property `ims.mapping.folder`.

See also *Deploying Server-side Mapping Files to the RPC Server*.

Deploying Server-side Mapping Files to the RPC Server

Deploy a server-side mapping file (Workbench file with extension `.svm`) to the RPC server manually. See *Server Mapping Files for COBOL* in the EntireX Workbench documentation.

» To deploy a server-side mapping file

- 1 Make sure the server-side mapping container (directory or folder) is configured. See *Server-side Mapping Files in the RPC Server*.
- 2 Copy the server-side mapping file to the server-side mapping container.

Undeploying Server-side Mapping Files from the RPC Server

Undeploy a server mapping file (Workbench file with extension .svm) from the RPC server manually. See *Server Mapping Files for COBOL*.

➤ **To undeploy a server-side mapping file manually**

- Delete the server-side mapping file from the server-side mapping container (directory or folder). See *Server Mapping Files for COBOL*.

Change Management of Server-side Mapping Files

Under UNIX and Windows, change management for a directory or folder (server-side mapping container, see *Server-side Mapping Files in the RPC Server*) is similar to change management within ordinary operating system directories (folders). All updates to the directory or folder done after a backup must be kept.

All EntireX Workbench server-side mapping files (.svm) added since the last backup should be available. See *Server Mapping Files for COBOL* in the EntireX Workbench documentation.

List Deployed Server-side Mapping Files

Use the Windows Explorer (for Windows) or the `ls` command (for UNIX) to list the contents of the server-side mapping container (directory or folder). See *Server-side Mapping Files in the RPC Server*.

Check if a Server-side Mapping File Revision has been Deployed

Server-side mapping files in the server-side mapping container correspond to EntireX Workbench files with extension .svm (same format). See *Server Mapping Files for COBOL* in the EntireX Workbench documentation. Each line relates to an IDL program and contains a creation timestamp at offset 276 (decimal) in the format `YYYYMMDDHHIISS`. Precision is 1/10 of a second. The creation timestamp can be checked.

The timestamp can be found on the same offset in the server-side mapping files stored in the server-side mapping container (directory or folder). See *Server-side Mapping Files in the RPC Server*.

Is There a Way to Smoothly Introduce Server-side Mapping Files?

All EntireX RPC servers can be executed without server-side mapping files. See [Server-side Mapping Files in the RPC Server](#). There is no need to install the server-side mapping container if the following conditions are met:

- You do not use features that require server mapping; see *When is a Server Mapping File Required?*
- Server-side type of COBOL mapping is switched on in the EntireX Workbench. If you have not used server-side mapping, we recommend you use client-side mapping. See *Server Mapping Files for COBOL*.

You can also call COBOL servers generated or extracted with previous versions of EntireX mixed with a COBOL server that requires server-side mapping. All EntireX RPC servers are backward compatible.

5 Scenarios

- COBOL Scenarios 22

COBOL Scenarios

Scenario I: Calling an Existing COBOL Server

» To call an existing COBOL server

- 1 Use the *IDL Extractor for COBOL* to extract the Software AG IDL and, depending on the complexity, also a server mapping file. See *When is a Server Mapping File Required?* in the EntireX Workbench documentation.
- 2 Build an EntireX RPC client using any EntireX wrapper. For a quick test you can:
 - use the IDL Tester; see *EntireX IDL Tester* in the EntireX Workbench documentation
 - generate an XML mapping file (XMM) and use the XML Tester for verification; see *EntireX XML Tester* in the XML/SOAP Wrapper documentation

See *Server Examples for z/OS IMS MPP* in the COBOL Wrapper documentation for COBOL RPC Server examples.