webMethods EntireX

IMS Connect RPC Server

Version 9.5 SP1

November 2013

This document applies to webMethods EntireX Version 9.5 SP1.

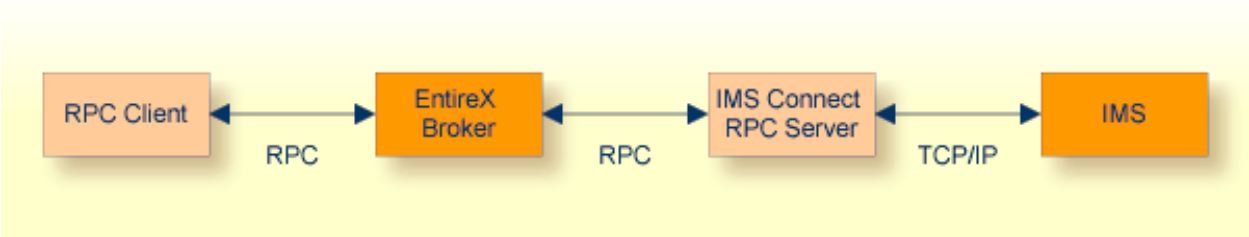
Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Document ID: EXX-IMSCONNECT-95SP1-20140628

Table of Contents

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 COBL 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

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

For the setup of the IMS Connect RPC Server there are

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

Location of the the IMS Connect RPC Server

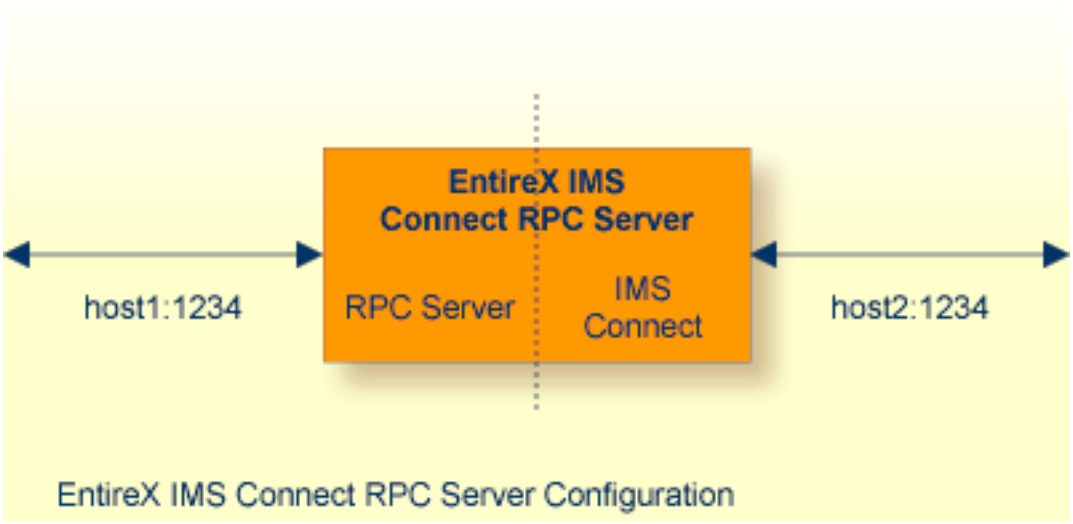
The IMS Connect RPC Server is contained in the file *entirex.jar*.

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.



EntireX IMS Connect RPC Server Configuration

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 connect to the same EntireX 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 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	Command-line Option	Default Value	Explanation
`entirex.bridge.verbose`		`no`	Verbose/trace mode of IMS Conne Server
`entirex.server.brokerid`	`-broker`	`localhost`	Broker ID
`entirex.server.serveraddress`	`-server`	`RPC/SRV1/CALLNAT`	Server address.
`entirex.server.userid`	`-user`	`IMSRPCServer`	The user ID for the Broker for RPC entirex.server.password.
`entirex.server.fixedservers`		`no`	no Use attach server to manage v threads. yes Run minimum number of ser threads.
`entirex.server.minservers`		`1`	Minimum number of server threa
`entirex.server.maxservers`		`32`	Maximum number of server threa
`entirex.server.restartcycles`	`-restartcycles`	`15`	Number of restart attempts if the not available. This can be used to IMS Connect RPC Server running Broker is down for a short time.

Name	Command-line Option	Default Value	Explanation
entirex.server.password	-password		The password for secured Broker. For Java 1.4 and above, th encrypted and written to entirex.server.passw To change the password, s password in the propertie *entirex.imsconnect.propertie* To disable password encry entirex.server.passw Default for this property i For Java 1.3 and below, th encryption is not available
entirex.server.properties	-propertyfile	entirex.server.properties	The file name of the prope
entirex.server.security	-security	no	no|yes|autol name of Br object
entirex.server.encryptionlevel		0	Encryption level. Valid va
entirex.server.compresslevel	-compresslevel	0	Permitted values (you can the numeric value) BEST_COMPRESSION BEST_SPEED DEFAULT_COMPRESSION DEFLATED NO_COMPRESSION N Y
entirex.server.waitattach		600S	Wait timeout for the attac
entirex.server.waitserver		300S	Wait timeout for the work
entirex.timeout		20	TCP/IP transport timeout. *Transport Timeout* under *W* *Applications - EntireX Java*
entirex.server.verbose	-verbose	no	Verbose output to standar
entirex.server.logfile	-logfile		Path and name of the trac
entirex.trace	-trace	0	Trace level (1,2,3).
entirex.server.monitorport	-smhport	0	The port where the server commands from the Syste Hub (SMH). If this port is and management by the S

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 EBCIDIC 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		The folder where the server expects SVM files (extension .svm). SVM files are generated by the IDL Extractor for COBOL and COBOL Wrapper. See *Server Mapping Deployment*.
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.

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=9.5.0.0`

3 Handling SVM Files

A server mapping file (SVM) enables the RPC server to correctly support special COBOL syntax such as `REDEFINE`s, `JUSTIFIED`, `SYNCHRONIZE` and `OCCURS DEPENDING ON` clauses, `LEVEL-88` fields, etc. If one of these elements is used, the EntireX Workbench automatically extracts an SVM file in addition to the IDL (interface definition language), or an SVM file is generated by the COBOL Wrapper for a server skeleton. The SVM file is used at runtime to marshal and unmarshal the RPC data stream.

SVM Files in the EntireX Workbench

In the *EntireX Workbench*, an SVM file has to relate to an appropriate IDL file. Therefore, you always have to keep the IDL file and the SVM file together in the same folder.

If there is an SVM file and a corresponding IDL file,

- at least one of the IDL programs in the corresponding IDL file requires server-mapping information to correctly call the target server. For those IDL programs, there is an SVM entry (line) in the Workbench SVM file.
- deployment of the SVM file to the RPC server is mandatory, see *Server Mapping Deployment*.

If there is an IDL file but no corresponding SVM file,

- there is no IDL program that requires server mapping information.

SVM Files in the RPC Server

For an IMS Connect RPC Server, copy the SVM files of the Workbench manually to a directory (folder) as operating system files. This directory (folder) is provided to the IMS Connect RPC Server with the property `ims.mapping.folder`. See ***Configuring the IMS Connect Side***.

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

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

See also *Deploying a Server Mapping File* in the COBOL Wrapper documentation.

Source Control of SVM Files

Because SVM entries within an SVM file contain text data only, a Workbench SVM file is text-based (although it is not intended for human consumption). Therefore, you can include it in your source control management together with the IDL file and the COBOL source(s) as a triplet that should always be kept in sync.

Change Management of SVM Files

For a IMS Connect RPC Server, change management of the SVM directory (see ***SVM Files in the RPC Server***) is similar to change management within ordinary operating system directories (folders). All updates to the SVM directory done after a backup must be kept.

All Workbench SVM files added since the last backup should be available.

Compare SVM Files

For SVM files in the *EntireX Workbench* format, you can use a third party file/text compare tool to check if two files are identical.

The SVM entries (corresponding to lines in a Workbench SVM file) contain a creation timestamp at offset 276 (decimal) in the format `YYYYMMDDHHIISST`. The precision is 1/10 of a second.

List Deployed SVM Files

To list the contents of an SVM directory (see ***SVM Files in the RPC Server***), use the Windows Explorer (for Windows) or the `ls` command (for UNIX).

Check if an SVM File Revision has been Deployed

SVM entries (corresponding to lines in Workbench SVM files) contain a creation timestamp at offset 276 (decimal) in the format `YYYYMMDDHHIISST`. Precision is 1/10 of a second. The creation timestamp can be checked.

The timestamp can be found on the same offset in the SVM files stored as operating system files in SVM directories.

When is an SVM File Required?

For the IDL Extractor for COBOL

Interface Type	COBOL Syntax	COBOL Mapping Editor	SVM Required	More Information
CICS with DFHCOMMAREA Calling Convention and `IN` different to `OUT`	all		yes	*CICS with `DFHCOMMAREA` Calling Convention* under *Introduction to the IDL Extractor for COBOL* | *CICS `DFHCOMMAREA`* under *COBOL Parameter Selection*
CICS Channel Container Calling Convention	all		yes	*CICS with Channel Container Calling Convention*
CICS with DFHCOMMAREA Large Buffer Interface	all		yes	*CICS with `DFHCOMMAREA` Large Buffer Interface*
IMS MPP Message Interface (IMS Connect)	all		yes	*IMS MPP Message Interface (IMS Connect)*
IMS BMP with Standard Linkage Calling Convention	all		yes	*IMS BMP with Standard Linkage Calling Convention*
Micro Focus with Standard Linkage Calling Convention	`BINARY` clause		yes	*Micro Focus with Standard Linkage Calling Convention*
all	`OCCURS DEPENDING ON` clause		yes	*Tables with Variable Size - `DEPENDING ON` Clause* under *COBOL to IDL Mapping* in the IDL Extractor for COBOL documentation
all	`REDEFINES` clause		yes	*`REDEFINE` Clause*
all	`TRAILING [SEPARATE]` clause		yes	*`SIGN LEADING` and `TRAILING SEPARATE` Clause*
all	`LEADING [SEPARATE]` clause		yes	*`SIGN LEADING` and `TRAILING SEPARATE` Clause*
all	`ALIGNED RIGHT` attribute		yes	
all	all	Rename of program	yes	*The Software AG IDL Tree Pane* under *Mapping Editor User Interface* in the IDL Extractor for COBOL documentation

Interface Type	COBOL Syntax	COBOL Mapping Editor	SVM Required	More Information
all	all	Map to operation	yes	*Context Menu* under *The COBOL Parameters Pane*
all	all	Map to constant	yes	*Context Menu*
all	all	Suppress	yes	*Context Menu*
other combinations			no	

For the COBOL Wrapper

This depends on the interface type chosen and the IDL type:

Interface Type	IDL Type	COBOL Wrapper	SVM Required	More Information
CICS with DFHCOMMAREA Large Buffer Interface	all		yes	*CICS with DFHCOMMAREA Large Buffer Interface* under *COBOL Server Interface Types*
CICS with Channel Container Calling Convention	all		yes	*CICS with Channel Container Calling Convention*
IMS BMP with Standard Linkage Calling Convention	all		yes	*IMS BMP with Standard Linkage Calling Convention*
Micro Focus	I2 or I4		yes	*Micro Focus with Standard Linkage Calling Convention* | *IDL Data Types* under *Software AG IDL File* in the IDL Editor documentation
all	IDL unbounded array		yes	`array-definition` under *Software AG IDL Grammar* in the *IDL Editor* documentation
all	IDL unbounded group		yes	`group-parameter-definition` under *Software AG IDL Grammar* in the *IDL Editor* documentation
all	all	IDL program name is not a valid COBOL name and is therefore adapted, or the COBOL program name is customized	yes	*Customize Automatically Generated Server Names*
other combinations			no	

Is There a Way to Smoothly Introduce SVM Files?

All EntireX RPC servers can be executed without SVM files. There is no need to install the SVM container (see ***SVM Files in the RPC Server***) as long as you do not use features that require SVM files (see ***When is an SVM File Required?***). You can also call COBOL servers generated or extracted with previous versions of EntireX mixed with a COBOL server that requires SVM files. All EntireX RPC servers are backward compatible.