webMethods EntireX

EntireX Java RPC Server

Version 9.12

October 2016

WEBMETHODS

Document ID: EXX-JAVARPC-912-20181116

Table of Contents

1 About this Documentation

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
`Monospace font`	Identifies service names and locations in the format *`folder.subfolder.service`*, APIs, Java classes, methods, properties.
`Italic`	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.

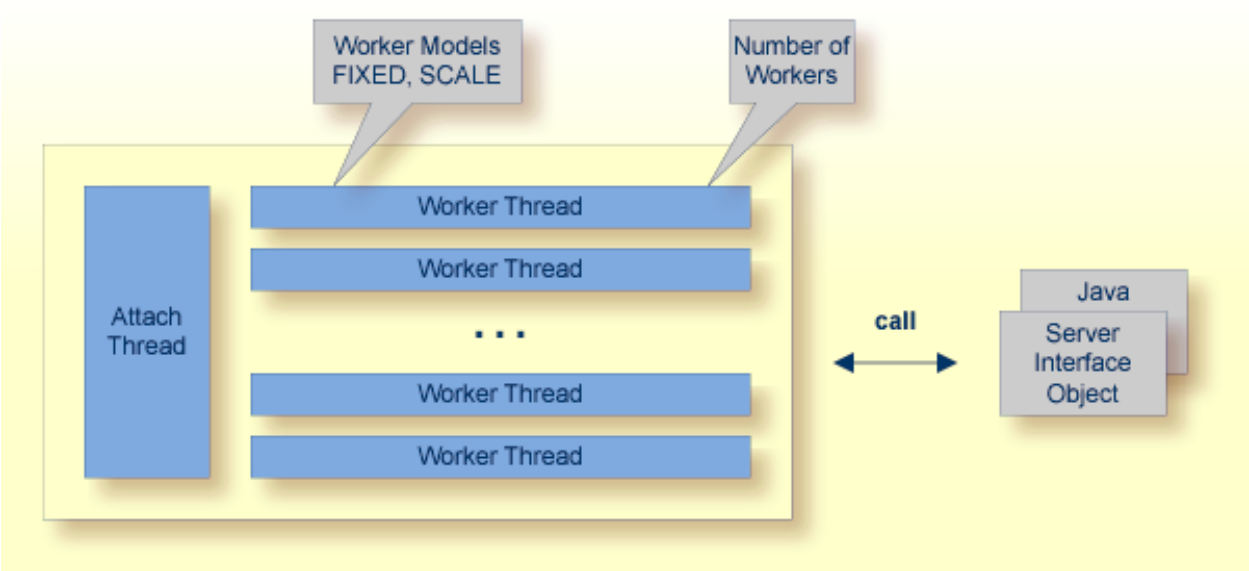
Data Protection

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 Java RPC Server

The EntireX Java RPC Server allows standard RPC clients to communicate with servers written in Java. It works together with the Java Wrapper and calls Java server interface objects.

Worker Models



RPC requests are worked off inside the RPC server in worker threads. Every RPC request occupies during its processing a worker thread. If you are using RPC conversations, each RPC conversation requires its own thread during the lifetime of the conversation. The Java RPC Server 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. The Java RPC Server provides two worker models:

The *fixed* model creates a fixed number of worker threads. If `entirex.server.fixedservers=yes`, the number of worker threads is specified in `entirex.server.minservers` is started and the server can process this number of parallel requests.

- **FIXED**
- `SCALE`
  The *scale* model creates worker threads depending on the incoming load of RPC requests. If `entirex.server.fixedservers=no`, the number of worker threads balances between what is specified in `entirex.server.minservers` and what is specified in `entirex.server.maxservers`. This is done by a so-called attach thread. At startup, the number of worker threads is the number specified in `entirex.server.minservers`. A new worker thread starts if the broker has more requests than there are worker threads waiting. If more than the number specified in `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`.

3 Administering the Java RPC Server

Customizing the RPC Server

The following elements are used for setting up the Java RPC Server:

- Start Script
- Properties File

Start Script

The start script for the Java RPC Server is called *jrpcserver.bsh* (UNIX) or *jrpcserver.bat* (Windows) and is provided in the *bin* folder of the installation directory. You may customize this file.

You can set the environment variable `JAVA_HOME` for the location of the Java interpreter. Set the classpath to entirex.jar and the path to the Java server interface objects.

The script files that start the Java RPC Server allow you to pass properties as command-line options as described in the table below. Alternatively, you can use system properties or a property file. The command-line option has the highest priority; the system property has second priority, and the entries of a property file have third priority.

Example:

```
java -Dentirex.server.properties=rpcserver.properties -classpath <entirex.jar with ↩
path>:<path to your server
interface objects> com.softwareag.entirex.aci.RPCServer
```

Properties File

The default name of the properties file is `entirex.server.properties`. The file is searched for in the directory of the ***Start Script***. It can be changed by assigning an arbitrary file name with a path to a property with the name `entirex.server.properties`.

A sample properties file is contained in subfolder *config* of the installation folder.

Configuring the RPC Server

Property Name	Command-line Option	Default	Explanation

`entirex.rpcserver.packagename.`*`library_name`*			The Java RPC Server car
			names if the package na
			`library-definition`
			server. For each library
			`entirex.rpcserver.`
			value of the package.
			
			Example for the library
			
			`entirex.rpcserver.`
			
			The library name must
			specified when the serv
			Properties under *Using t*

`entirex.server.brokerid`	`-broker`	`localhost`	Broker ID
`entirex.server.` `codepage`	`-codepage`		EntireX Java componen Java virtual machine (J representation within J encoding sent to or rece encoding is also transfe the broker to tell the bro on the internationalizat string) is used by the br internationalization app codepage (locale string) ■ follow the rules descr ■ be a codepage suppo ■ be a codepage suppo encodings, see *Suppo* documentation. If these rules are not obs With the codepage para used to convert the Uni Java to the multibyte or from the broker without of the JVM. Changing the default er the encoding for termin undesired. For how to c

Property Name	Command-line Option	Default	Explanation
			see your JVM documentation. C it can be changed with the `file`

`entirex.server.`
`compresslevel`
	`-compresslevel`	`0` (no compression)	Permitted values (you can enter
			`BEST_COMPRESSION`
			`BEST_SPEED`
			`DEFAULT_COMPRESSION`
			`DEFLATED`
			`NO_COMPRESSION`
			`N`
			`Y`

`entirex.server.` `customclass`	`-customclass`		This class is used for custom init server. In addition, this class all conversation and handling the t See `ServerImplementation` fo
`entirex.server.` `environment`			Can be used in a user-written tra `BrokerService`, `setEnvironm` (EntireX Java ACI).
`entirex.server.` `fixedservers`		`no`	If `no`, use attach thread to mana Otherwise run a fixed number o
	`-help`		Display usage of the command-
`entirex.server.` `logfile`	`-logfile`		Path and name of the trace outp in the name are resolved only if
`entirex.server.` `maxservers`		`32`	Maximum number of worker th
`entirex.server.` `minservers`		`1`	Minimum number of server thr
`entirex.server.` `name`			The name of the server.

`entirex.server.`
`password`
	`-password`	`yes`	The password for secured acces
			is encrypted and written to the
			`entirex.server.password.`
			To change the password, set the
			file (default is `entirex.serve`
			To disable password encryption
			`entirex.server.passworden`

`entirex.server.` `properties`	`-propertyfile`	`entirex.server.` `properties`	The name of the property file.
`entirex.server.` `restartcycles`	`-restartcycles`	`15`	Number of restart attempts if th can be used to keep the Java RP Broker is down for a short time

Property Name	Command-line Option	Default	Explanation
`entirex.server.security`	`-security`	`no`	`no`/`yes`/`auto`/Name of
`entirex.server.serveraddress`	`-server`	`RPC/SRV1/CALLNAT`	Server address
`entirex.server.serverlog`	`-serverlog`		Name of the file where logged. Used by the Wi
`entirex.server.userid`	`-user`	`JavaServer`	The user ID for the Brok `entirex.server.pas`
`entirex.server.verbose`	`-verbose`	`no`	Verbose output to stanc
`entirex.server.waitattach`		`600S`	Wait timeout for the att
`entirex.server.waitserver`		`300S`	Wait timeout for the wo
`entirex.timeout`		`20`	TCP/IP transport timeo under *Writing Advancea*
`entirex.trace`	`-trace`	`0`	Trace level (`1`,`2`,`3`).

Using SSL/TLS with the RPC Server

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 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 Java 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 SSL Clients*.

3 Make sure the SSL server to which the Java RPC Server 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
- *Settting up and Administering the EntireX Broker SSL Agent* in the UNIX and Windows Administration documentation

Starting the RPC Server

To start the Java RPC Server

- Use the ***Start Script***.

  Or:

  At the command prompt, enter:

```
java com.softwareag.entirex.aci.RPCServer
```

You can pass command-line options and customize your environment as described under ***Start Script***.

Stopping the RPC Server

To stop the Java RPC Server

- Use the command `stopServer`. See Stop EntireX Broker Server Instances in Command Central's Command-line Interface.

  Or:

  Stop the server instance using Command Central's Graphical User Interface. See *Stopping a Server Instance*.

  Or:

  Use the command `stopService`. See Stop Running EntireX Broker Services in Command Central's Command-line Interface.

  Or:

  Stop the service using Command Central's Graphical User Interface. See *Stopping a Service*.

  Or:

  Use the command-line utility `etbcmd`. See `etbcmd` under *Broker Command-line Utilities* in the platform-specific Administration documentation.

  Or:

  Use `CTRL-C` in the session where you started the RPC server instance.

  Or:

  Under UNIX, enter command `kill -process-id`.

Running the RPC Server as a Windows Service

For general information see *Running an EntireX RPC Server as a Windows Service*.

To run the Java RPC Server as a Windows Service

1 Customize the ***Start Script*** according to your system installation.

Note: The script file must pass external parameters to the RPC server and use the reduced signalling of the JVM (option `-Xrs`):

```
java -Xrs com.softwareag.entirex.aci.RPCServer %*
```

If `-Xrs` is not used, the JVM stops and an entry 10164002 is written to the event log when the user logs off from Windows.

See also ***Starting the RPC Server***.

2 Test your RPC server to see whether it will start if you run your script file.

3 Use the *EntireX RPC Service Tool* and install the `RPCService` with some meaningful extension, for example `MyServer`. If your ***Start Script*** is *jrpcserver.bat*, the command will be

```
RPCService -install -ext MyServer -script install_path\EntireX\bin\jrpcserver.bat
```

The log file will be called *RPCservice_MyServer.log*.

4 In **Windows Services** menu (**Control Panel** > **Administrative Tools** > **Services**) select the service: `Software AG EntireX RPC Service [MyServer]` and change the property `Startup Type` from "Manual" to "Automatic".

Application Identification

The application identification is sent from the 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 Java RPC Server these values are:

Identification Part	Value
Application name	`ANAME=Java RPC Server`
Node name	`ANODE=`*`host_name`*
Application type	`ATYPE=Java`
Version	`AVERS=9.12.0.0`

4 Scenarios and Programmer Information

Writing a New Java Server

To write a new Java server

1. Use the Java Wrapper to generate a Java server interface object. See *Generating a Java Server Interface Object*. Write your Java server and proceed as described under *Generating a Java Server Interface Object*.
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