Administering the EntireX XML/SOAP RPC Server

With the XML/SOAP RPC Server you can process XML-based server calls from EntireX RPC clients/Natural RPC clients. The EntireX RPC client communicates with the XML-based server, using the XML/SOAP RPC Server.

Administering the EntireX XML/SOAP RPC Server

The XML/SOAP RPC Server uses the following, in the following order of priority:

1. **Command-line Parameters**
   The command-line parameters have the highest priority.

2. **Properties File**
   The properties file is located in the working directory by default. It should define parser settings and the location of the configuration file. The default name of the properties file is *entirex.xmlrpcserver.properties*. Furthermore it may contain several properties for the server (see the table below).

3. **Configuration File**
   The configuration file (XML format) has the lowest priority. It contains a list of target servers, including the mapping file associated with them and may contain information about the broker if not already given in the command-line or property file.

If the properties file does not specify the location and name of the configuration file, the configuration file in the working directory is used.

Additionally, Java System properties are available to administer the XML/SOAP RPC Server. These properties are independent of the administration possibilities listed above.

Java System Property	Description	Values	Default
`http.keepAlive`	Enable/disable HTTP persistence	true, false	true
`http.maxConnections`	Define the maximum number of HTTP connection to a host. **Note:** Requires `http.keepAlive=true`	Integer > 0	5

Command-line Parameters

Name	Command-line Option	Default Value	Explanation
`entirex.server.brokerid`	`-broker`	`localhost`	Broker ID
`entirex.server.codepage`	`-codepage`		The codepage the server uses. Permitted values are the names of the codepages the JVM supports. Use the value LOCAL when the default codepage of the JVM should be used. See *Using Internationalization with EntireX XML Components* for details.
`entirex.server.compresslevel`	`-compresslevel`	0 (no compression)	Permitted values (you can enter the text or the numeric value): `BEST_COMPRESSION` 9 `BEST_SPEED` 1 `DEFAULT_COMPRESSION` -1, mapped to 8 `DEFLATED` 8 `NO_COMPRESSION` 0 N 0 Y 8
`entirex.server.development.relativepaths`		`false`	The file locations of deployed XMM and WSDL files are written as relative paths in configuration file of the XML/SOAP RPC Server.
`entirex.server.environment`			Can be used in a user-written translation exit of the Broker. See `BrokerService`, `setEnvironment(java.lang.String)` (EntireX Java ACI).
`entirex.server.fixedservers`		no	If no, use attach server to manage worker threads, otherwise run minimum number of server threads.
`entirex.server.ignoreSOAPActionNamespace`		`false`	`true`: Only the name part of `SOAPAction` is used, the namespace is ignored. `false`: The `SOAPAction` value is used as defined. **Note:** If a WSDL file is configured for this method, the property will be ignored and `SOAPAction` value defined in WSDL is used.
`entirex.server.logfile`	`-logfile`		Name of the log file, default is standard output. Environment variables in the name are resolved only if used as a command-line option.
`entirex.server.maxservers`		32	Maximum number of worker threads.
`entirex.server.minservers`		1	Minimum number of server threads.
`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 the management by the SMH is disabled.
`entirex.server.name`			The name of the server.

Name	Command-line Option	Default Value	Explanation
entirex.server.password	-password		The password for secured access to the Broker. The password is encrypted and written to the property entirex.server.password.e. To change the password, set the new password in the properties file (default is entirex.server.properties). To disable password encryption, set entirex.server.passwordencrypt=no. Default for this property is yes.
entirex.sdk.xml.runtime.propertyfile	-propertyfile	entirex.xmlrpcserver.properties	The file name of the property file.
entirex.sdk.xml.runtime.configurationfile	-configurationfile	entirex.xmlrpcserver.configuration.xml	Location and name of configuration file.
entirex.server.restartcycles	-restartcycles	15	Number of restart attempts if the Broker is not available. This can be used to keep the Java RPC Server running while the Broker is down for a short time.
entirex.server.security	-security	no	no/yes/auto/Name of BrokerSecurity object.
entirex.server.serveraddress	-server	RPC/SRV1/CALLNAT	Server address.
entirex.server.serverlog	-serverlog		Name of the file where start and stop of worker threads is logged. Used by the Windows RPC Service.
entirex.server.userid	-user	JavaServer	The user ID for the Broker for RPC. See entirex.server.password .
entirex.server.verbose	-verbose	no	Enable verbose output to the log file.
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 *Setting the Transport Timeout* under *Writing Advanced Applications - EntireX Java ACI*.
entirex.trace	-trace	0	Trace level (1,2,3).
entirex.sdk.xml.runtime.xmlparserfactory	-jaxp.saxparserfactory	com.ctc.wstx.stax.WstxInputFactory	Location and name of stream parser factory class.
entirex.sdk.xml.runtime.useCharacterReference		no	Enables or disables the usage of character references. Defined value = yes,no.
entirex.sdk.xml.runtime. defaultFaultDocumentFormat		soap	Define the protocol used for fault document generation if no fault document is defined. Defined values = soap, xml.

Sample Properties File

The following is a sample properties file *entirex.xmlrpcserver.properties*:

```
# Example server configuration
#
# parameter for xml stream parser
entirex.sdk.xml.runtime.xmlparserfactory=com.ctc.wstx.stax.WstxInputFactory
# xmlruntime configuration file
entirex.sdk.xml.runtime.configurationfile=entirex.xmlrpcserver.configuration.xml
#
# Basic properties
entirex.server.brokerid=localhost
entirex.server.serveraddress=RPC/XMLSERVER/CALLNAT
entirex.server.userid=XMLRPCServer
```

Configuration File for the XML/SOAP RPC Server

- Introduction
- Sample Configuration File
- TargetServer Block

Introduction

The configuration file for the EntireX XML/SOAP RPC Server is written in XML format.

The document frame is:

```
<?xml version="1.0" encoding="UTF-8" ?>
<EntireX xmlns="http://namespaces.softwareag.com/entirex/xml/runtime/configuration" version="7.2.1" >
      <XmlRuntime Version="1">
      <!-- information for XML/SOAP RPC Server-->
      </XmlRuntime>
</EntireX>
```

The default name of the configuration file is *entirex.xmlrpcserver.configuration.xml*.

The XMLRPCServer information contains two information blocks, one for the EntireX Broker information and one for a list of target servers.

Sample Configuration File

```
<?xml version="1.0" encoding="UTF-8" ?>
<EntireX
  xmlns="http://namespaces.softwareag.com/entirex/xml/runtime/configuration"
  version="8.0">
  <XmlRuntime Version="1">

   <TargetServer name="http://localhost:1973/MyService">
      <xmms>
        <exx-xmm name="c:\mydir\xmmfiles\XmmExample.xmm"
          soapVersion="1.1"
          wsdl="c:/mywsdl.wsdl" service="myservice"
          port="myserviceSOAP11Port" repository="c:\myrepository"\>
      </xmms>
    </TargetServer>
  </XmlRuntime>
<EntireX>
```

TargetServer Block

The section <TargetServer>

- specifies a Web service address (currently only http(s) is possible)
- contains the IDL-XML mapping files (XMM)
- allows specification of basic authentication with a fixed user/password within the tag <TargetServer>:

Attribute	Req/ Opt	Description
`basicAuthentication`	O	`true` Activate the basic authentication. If attributes `user` and `password` are set, these credentials are used for basic authentication. Otherwise the current credentials of the calling client are used. To set the basic authentication credentials on client side, the Natural logon must be enabled. User-specific credentials can be overwritten by setting RPC user ID and RPC password in the client application. `false` Deactivate basic authentication. All other parameters in this table are ignored.
`user`	O	Name of default user for basic authentication.
`password`	O	Password of default user for basic authentication.
`password-encryption`	O	Specifies how the password is encrypted. Possible values: `plainText` Default. `base64` `encrypt` The XML/SOAP RPC Server encrypts the password and sets this value.
`httpConnectionTimeout`	R	HTTP connection timeout in seconds.

See *Reference - HTTP and Java Interface* for explanation of attributes.

The section <xmm> contains the optional attributes for SOAP mapping.

Attribute	Description
`soapVersion`	Specifies a SOAP version: 1.1 (default) or SOAP 1.2.
`wsdl`	The location of WSDL file, using a WSDL file the target address is retrieved from WSDL file.
`service`	The service name in WSDL file.
`port`	The port name in WSDL file.
`repository`	The repository directory used for WS-* features. See Software AG Common Web Services Stack client repository.
`usernameToken`	Valid values: `PasswordText` | `PasswordDigest`. Prerequisites: Attribute `repository` must be defined and module `rampart` must be engaged. See also *XML/SOAP RPC Server with UsernameToken*.

The list of target servers (based on the target server entries starting with tag `TargetServer` and have a mandatory HTTP address) is assigned to the attribute name. Each `TargetServer` entry can have a list of XMMs for this server.

Caution:
It is not allowed to use one XMM in more than one `TargetServer` entry inside one configuration file. Using different XMMs with a common definition results in unexpected behavior of XML/SOAP RPC Server.

Configuring the XML/SOAP RPC Server

To configure the XML/SOAP RPC Server

1. Specify the file *entirex.xmlrpcserver.properties* in the directory where the XML/SOAP RPC Server is started.
2. Specify the JAXP parameters. This step is optional if these parameters are already specified in your environment.
3. Specify the location of the configuration file.
4. Specify the configuration file: *entirex.xmlrpcserver.configuration.xml*.
5. For specifying features such as WS-Policy, see also configuration of Software AG Common Web Services Stack.

Tip:
If you are using the XML/SOAP RPC Server with an HTTP server located outside the firewall, set the following Java properties:

- `http.proxyHost`
- `http.proxyPort`

- `https.proxyHost`
- `https.proxyPort`
- `http.nonProxyHosts`
- `https.nonProxyHosts`
- `http.proxyUser`
- `https.proxyUser`
- `http.proxyPassword`
- `https.proxyPassword`

XML/SOAP RPC Server with HTTP Basic Authentication

The XML/SOAP RPC Server uses basic authentication for a Web service if the configuration contains the attribute `basicAuthentication` block in `<TargetServer>`. Basic authentication is used for all calls associated with defined XMM files for the `<TargetServer>`.

Basic authentication can be used with fixed credentials or credentials set from the client application:

- If `<TargetServer>` contains attributes `user` and `password`, these settings are used for basic authentication.
- Otherwise the client application must provide the credentials: Enable Natural logon and set RPC user ID and RPC password.

See *Configuration File for the XML/SOAP RPC Server*.

XML/SOAP RPC Server with UsernameToken

The XML/SOAP RPC Server uses UsernameToken security for a Web service if the configuration contains the attribute `usernameToken` in `<xmm>`. The XML/SOAP RPC Server supports two kinds of UsernameToken:

- PasswordText
- PasswordDigest

The XML/SOAP RPC Server configuration must define the repository, for example:

```
<exx-xmm name="AService.xmm" soapVersion="1.1"
repository="myrepository" usernameToken="PasswordText" />
```

The repository must contain module `rampart`. In the configuration file (axis2.xml) the rampart module must be engaged (`<module ref="rampart"/>`) and the phase `PreSecurity` can be empty (`<phase name="PreSecurity" />`).

In the client application, the Natural logon must be set. Additionally the client application should set RPC user ID and RPC password.

See *Configuration File for the XML/SOAP RPC Server*.

Using SSL or TLS with the XML/SOAP RPC Server

Using HTTPS with XML/SOAP RPC Server requires setting Java properties and changing the protocol from http to https in the configuration file. This section covers the following topics:

- SSL or TLS Settings
- Sample Start Script
- Configuration File Settings

See also *Configuration File for the XML/SOAP RPC Server*.

SSL or TLS Settings

To configure SSL communication for the JRE

- Set the following properties:
  - **-Djavax.net.ssl.keyStore=<filename-without-blanks>**
    Here we keep the certificate and the private signing key of our client application, which is the EntireX XML/SOAP RPC Server.
  - **-Djavax.net.ssl.keyStorePassword=<you-should-know-it>**
    The password that protects the keystore.
  - **-Djavax.net.ssl.keyStoreType=pkcs12**
    If not jks (default).
  - **-Djavax.net.ssl.trustStore=<filename-without-blanks>**
    Here we keep the trusted certificate of the Web service host or the certificate of its signing (issuing) certificate authority.
  - **-Djavax.net.ssl.trustStorePassword=<you-should-know-it>**
    The password that protects the truststore.
  - **-Djavax.net.ssl.trustStoreType=**
    If not jks (default).

For more information about Java and SSL, see your Java documentation (JSSE documentation).

Sample Start Script

```
set CLASSPATH=.;.\classes\entirex.jar;..\WS-Stack\lib\wsstack-client.jar

set PROXYSETTINGS=-Dhttps.proxySet=true
-Dhttps.proxyHost=sslproxy.mydomain
-Dhttps.proxyPort=443
-Dhttps.nonProxyHosts="localhost"
```

```
set SSL=-Djavax.net.ssl.keyStore=C:\myKeystore.p12
-Djavax.net.ssl.keyStorePassword=myKeystorePassword
-Djavax.net.ssl.keyStoreType=pkcs12
-Djavax.net.ssl.trustStore=C:\myTrustStore.jks
-Djavax.net.ssl.trustStorePassword=myTruststorePassword

java -classpath %CLASSPATH% %SSL% %PROXYSETTING% com.softwareag.entirex.xml.rt.XMLRPCServer
```

For the changes that are required to the start script, see your Java documentation (JSSE documentation).

Configuration File Settings

Specify the fully qualified host name as TargetServer. The host name has to match the CN (Common Name) item of the host certificate.

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<EntireX
xmlns="http://namespaces.softwareag.com/entirex/xml/runtime/configuration" version="8.0"
>
    <XmlRuntime Version="1">
       <TargetServer name="https://targethost:8080/entirex/xmlrt">
            <xmms>
                <exx-xmm name="yourFile1.xmm" />
                <exx-xmm name="yourFile2.xmm" />
            </xmms>
        </TargetServer>
    </XmlRuntime>
</EntireX>
```

Java API for XML/SOAP RPC Server

The Java API for XML/SOAP RPC Server is a functional extension to the XML/SOAP RPC Server. It allows you to direct the calls to a Java object instead of a Web service (via HTTP(s)). The usage of Java API for XML/SOAP RPC Server is similar to what is known for the XML/SOAP RPC Server. It only differs in the start script and a new (additional) keyword in the configuration file. See *Configuring the XML/SOAP RPC Server* above.

- Properties File
- Configuration File
- Implementation of the Java API for XML/SOAP RPC Server
- Start Script

Properties File

The property file is the same as the *Sample Properties File* for the XML/SOAP RPC Server.

Configuration File

The Java API for XML/SOAP RPC Server also uses the same configuration file as the XML/SOAP RPC Server.

The services (programs) directed to the Java interface of the XML/SOAP RPC Server have to use a special keyword "xmlrpcServerClass" as the value of the attribute "Targetserver". A mixture of targetserver with Java and http-interface is also possible.

Example:

```
<?xml version="1.0" encoding="UTF-8" ?>
<EntireX
xmlns="http://namespaces.softwareag.com/entirex/xml/runtime/configuration"
version="8.3"
>
<XmlRuntime Version="1">
<BrokerInfo>
<BrokerId>localhost:1971</BrokerId>
<ServerAddress>RPC/SRV1/CALLNAT</ServerAddress>
</BrokerInfo>
<TargetServer name="xmlrpcServerClass">
<xmms>
<exx-xmm name="java-service1.xmm" />
<exx-xmm name="java-service2.xmm" />
<exx-xmm name="java-service3.xmm" />
</xmms>
</TargetServer>
<TargetServer name="http://myWebService">
<xmms>
<exx-xmm name="http-service1.xmm" />
<exx-xmm name="http-service2.xmm" />
</xmms>
</TargetServer>
</XmlRuntime>
</EntireX>
```

Implementation of the Java API for XML/SOAP RPC Server

The Java API for XML/SOAP RPC Server requires a user-written Java class initializing the XML/SOAP RPC Server and implementing the XMLRPCServerInterface.

Example:

```
import java.util.Properties;
import com.softwareag.entirex.xml.rt.XMLRPCServerInterface;
import com.softwareag.entirex.xml.rt.XMLRPCServer;
public class MyXMLRPCServer implements XMLRPCServerInterface
{
  public MyXMLRPCServer ()
  {
    XMLRPCServer xmlRpcServer = new XMLRPCServer();
    // register your implementation of XMLRPCServerInterface
    xmlRpcServer. registerXMLRPCServerClass ((XMLRPCServerInterface) this);
    // start XML/SOAP RPC Server with arguments (same as command line)
    xmlRpcServer.start(new String[0]);
  }

  // mandatory method invoke (from XMLRPCServerInterface)
  // - thread synchronization must be done by application if required
  // - properties object contains property "charset" (as used in xml-declaration)
  // and property "java.charset" - the corresponding Java codepage
  // - Exception thrown from this method is mapped to error class 2000 and error number 200,
  //   with exception information in errortext

  public byte[] invoke(byte[] requestDocument, Properties properties)
    throws Exception
  {
    byte[] response = null;
    // TODO <insert application code here>
    return response;
  }

  public static void main(String[] args)
  {
    MyXMLRPCServer myServer = new MyXMLRPCServer ();
  }
}
```

Start Script

The XML/SOAP RPC Server with Java interface must be started by implementing XMLRPCServerInterface as in this example:

```
java  -classpath "%PARSER%;%CLASSPATH%" MyXMLRPCServer
```

Starting the XML/SOAP RPC Server

To start the XML/SOAP RPC Server

- Use the shell script *jxmlrpcserver* in the subfolder *bin* of the installation directory.

  Or:
  At the command prompt, enter:

```
java com.softwareag.entirex.xml.rt.XMLRPCServer
```

If the Java interpreter is not called "java", change the call to "java".

- 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 generated proxies.
- The XML/SOAP RPC Server accepts two unnamed parameters, the Broker ID and the server address. Default values are `localhost:1971` and `RPC/SRV1/CALLNAT`.

Stopping the XML/SOAP RPC Server

To stop the XML/SOAP RPC Server

- Use the function Deregister a Service or Deregister a Server of the System Management Hub. This method ensures that the deregistration from the Broker is correct.

Running the XML/SOAP RPC Server as a Windows Service

For general information, see *Administering the EntireX RPC Server* or for information on installing and tracing the Windows service, see *Running an EntireX RPC Server as a Windows Service*.

To run the XML/SOAP RPC Server as a Windows Service

1. Customize *jxmlrpcserver.bat* (or any other script file) according to your system installation. See *Running the XML/SOAP RPC Server as a Windows Service*.

   Note:
   The script file must pass external parameters to the RPC server:
   ```
   java com.softwareag.entirex.xml.rt.XMLRPCServer BrokerId
   ServerAddress %*
   ```

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

3. Install RPCService with some meaningful extension, for example:

   ```
   RPCService -install -ext <ext> -script
   "C:\SoftwareAG\EntireX\bin\jxmlrpcserver.bat"
   ```

4. In Windows Services menu (**Control Panel** > **Administrative Tools** > **Services**) select the service: Software AG EntireX RPC Service [<*ext*>]

   and change the property `Startup Type` from "Manual" to "Automatic".

Running the XML/SOAP RPC Server in the Software AG Runtime

This section covers the following topics:

- Introduction
- Configuration
- Deactivating an XML/SOAP RPC Server Permanently
- Starting and Stopping the XML/SOAP RPC Server using JMX (Java Management Extensions)
- Starting and Stopping the XML/SOAP RPC Server under Windows

See also *XML/SOAP RPC Server in the Software AG Runtime* under *Frequently Asked Questions (FAQ) and Troubleshooting* in the XML/SOAP Wrapper documentation.

Introduction

The Software AG Common Platform is a Java runtime environment based on the OSGi framework. It provides a standard platform on which to run Software AG products and the enterprise applications you develop around those products. The Software AG Common Platform provides common infrastructure for user authentication, event handling, and the execution of Web applications. Infrastructure components that the Software AG Common Platform provide include Software AG Security Infrastructure, Software AG Web Server based on Apache Tomcat, and Web Services Stack.

The Software AG Runtime is an installable instance of the Software AG Common Platform that functions as a stand-alone Tomcat server and a container for Web applications. EntireX uses the Software AG Runtime to host the EntireX XML/SOAP Listener and XML/SOAP RPC Server.

The Software AG Web Server based on Apache Tomcat is one of the basic infrastructure components provided by the Software AG Common Platform. It provides HTTP/HTTPS services, a JSP engine, and a servlet container. Unlike a typical Tomcat implementation, the Software AG Web Server is OSGi-based and supports both .WAR-based and .WAB-based web applications.

During startup, the Software AG Web Server (service name: Software AG Runtime), including the EntireX bundle, looks in the EntireX profile for file *<Installation home>/EntireX/etc/EXX/workspace/entirex.servers.properties*. This file defines an XML/SOAP RPC Server as within *entirex.xmlrpcserver.properties* and *entirex.xmlrpcserver.configuration.xml* located in the EntireX installation in subdirectory *config* by default.

Configuration

The file *entirex.servers.properties* defines the servers to be started. It is only read during startup of the Software AG Runtime. Set the following properties for each defined server:

Property Name	Description
server.<n>.kind	Must be "XMLRPCServer".
server.<n>.propertiesFile	Path to properties file (Java notation).
server.<n>.configurationFile	Path to configuration file (Java notation).

where <n> is a number identifying the server

Example of *entirex.servers.properties*:

```
server.1.kind=XMLRPCServer
server.1.propertiesFile=c:/SoftwareAG/EntireX/config/entirex.xmlrpcserver.properties
server.1.configurationFile=c:/SoftwareAG/EntireX/config/entirex.xmlrpcserver.configuration.xml
server.2.kind=XMLRPCServer
server.2.propertiesFile=c:/SoftwareAG/EntireX/config/entirex.myxmlrpcserver.properties
server.2.configurationFile=c:/SoftwareAG/EntireX/config/entirex.myxmlrpcserver.configuration.xml
```

Deactivating an XML/SOAP RPC Server Permanently

To stop any XML/SOAP RPC Server permanently (including the default XML/SOAP RPC Server), rename the configuration file `entirex.servers.properties` under *EntireX\etc\exx\workspace*, for example to `entirex.servers.properties.bak`.

Starting and Stopping the XML/SOAP RPC Server using JMX (Java Management Extensions)

To start and stop an XML/SOAP RPC Server, open a JMX tool, for example the Java Monitoring and Management Console (`jconsole`), located in the Java *bin* directory (sample path: *C:\SoftwareAG\jvm\w64_160\bin\jconsole.exe*). The tool should be connected to the Software AG Runtime JMX port remotely. The default number of this port is 8044 and is defined in *<Installation home>/ profiles/CTP/configuration/config.ini*.

Switch to tab MBeans and select item *com.softwareag.entirex.runtime.rpcserver*. The following operations are available:

Operation	Description
startServer	To start a registered and non-running XML/SOAP RPC Server. The parameter is the service name (e.g. RPC/XMLSERVER/CALLNAT).
stopServer	To stop a running XML/SOAP RPC Server. The parameter is the service name (e.g. RPC/XMLSERVER/CALLNAT).
registeredServer	Returns the list of service names of all configured XML/SOAP RPC Servers.
runningServer	Returns the list of service names of running configured XML/SOAP RPC Servers.
nonRunningServer	Returns the list of service names of non-running configured XML/SOAP RPC Servers.

Starting and Stopping the XML/SOAP RPC Server under Windows

Under Windows, the Software AG Runtime runs as a service and can be started and stopped using the Windows Administrative tools.

System Service

The relevant CTP profile has a Windows service named **Software AG Runtime**. You can see it in the Service dialog from the Windows Control Center. You can start/stop the service from this dialog.

Batch Scripts

The directory '<Installation Home>\ profile\CTP\bin' contains the following scripts to start and stop the Software AG Runtime:

Script	Description
start_runtime.bat	batch script for starting a profile's runtime. If it is an installed service for the profile, it will start the service by default.
stop_runtime.bat	batch script for stopping a profile's runtime. If it is an installed service for the profile and this service has been started, this script will stop the service.