software AG

webMethods EntireX

Listener for XML/SOAP

Version 10.8

October 2022

WEBMETHODS

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

Product Documentation

You can find the product documentation on our documentation website at **https://documentation.softwareag.com**.

In addition, you can also access the cloud product documentation via **https://www.softwareag.cloud**. Navigate to the desired product and then, depending on your solution, go to "Developer Center", "User Center" or "Documentation".

Product Training

You can find helpful product training material on our Learning Portal at **https://knowledge.softwareag.com**.

Tech Community

You can collaborate with Software AG experts on our Tech Community website at **https://tech-community.softwareag.com**. From here you can, for example:

- Browse through our vast knowledge base.

- Ask questions and find answers in our discussion forums.

- Get the latest Software AG news and announcements.

- Explore our communities.

- Go to our public GitHub and Docker repositories at **https://github.com/softwareag** and **https://hub.docker.com/publishers/softwareag** and discover additional Software AG resources.

Product Support

Support for Software AG products is provided to licensed customers via our Empower Portal at **https://empower.softwareag.com**. Many services on this portal require that you have an account. If you do not yet have one, you can request it at **https://empower.softwareag.com/register**. Once you have an account, you can, for example:

- Download products, updates and fixes.

- Search the Knowledge Center for technical information and tips.

- Subscribe to early warnings and critical alerts.

- Open and update support incidents.

- Add product feature requests.

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 **Listener for XML/SOAP**

The EntireX Listener for XML/SOAP runs in a servlet-enabled Web server with an installation of the Software AG Web Services Stack. It plugs the Web service's archives (Designer file with extension .aar) into Web servers. Typically the Web service clients access the Listener for XML/SOAP and send/receive XML/SOAP documents using HTTP/HTTPS.

See also Web Services Stack documentation in the *Software AG Infrastructure Administrator's Guide*, also available under *http://documentation.softwareag.com* > *Guides for Tools Shared by Software AG Products*.

Configuring Web Services

- Configuring inside a Web Service Archive
- Using an External Configuration File
- Using the Broker and RPC User ID/Password
- Logon to Natural Library

Configuring inside a Web Service Archive

Individual settings for Web services or groups of Web services are contained in the *services.xml* file that is part of a Web service archive (Designer file with extension .aar). Those settings originate from the Web Services Wrapper (default settings from generation) or from the *Web Services Stack Configuration Editor* (customized settings).

Using an External Configuration File

With an external configuration file you can override settings of the Web service's archive (Designer file with extension .aar) without modifying the Web service archive itself. This means you can use the same Web service archive in different Web server environments.

> **To use an external configuration file**

1 Define a name and a location for the external configuration file.

   In the parameter section of file *axis2.xml*, define a parameter `"EntireX-XML-Listener"` within a parameter `"services"`. For the attribute `"location"` in parameter `"services"`, specify an absolute or relative path to the external configuration file. File *axis2.xml* can be found in the *conf* directory or folder of the Web Services Stack Web application.

```
<parameter name="EntireX-XML-Listener">
   <parameter name="services" location="<path for file
                overwriting settings of EntireX services>" />
</parameter>
```

> **Notes:**

1. The path separator is a slash.

2. If you are using the Software AG Web Server based on Apache Tomcat, the location of *axis2.xml* is defined in configuration file *config.ini*, which is located in <SuiteInstallDir>/profiles/CTP/configuration by default. The default location of *axis2.xml* is *<SuiteInstallDir>/profiles/CTP/workspace/wsstack/repository/conf*. See also *Configuration > Web Services Stack Runtime > Runtime Configuration* in the Web Services Stack documentation in the *Software AG Infrastructure Administrator's Guide*, also available under *http://documentation.softwareag.com > Guides for Tools Shared by Software AG Products*.

3. The value of the location can contain operating system variables.

2   Define services in the external configuration file.

External configuration files are XML documents with a root element "serviceGroup". A service group is defined as a sequence of one or more services.

To identify the service you are defining, specify an identifier for the attribute "name", for instance <service name= "service100">.

To make common settings, that is, settings for all services, use an asterisk as identifier (<service name= "*">. Note that an individual setting can override a common setting.

EntireX web service parameters that can be set are defined as subelements of "service". See the table below.

Parameter	Description	More Information
<exx-brokerID>	The broker ID to use.	
<exx-service>	The service name is the triplet of server class/server name/service.	
<exx-userID>	The user ID for access to the broker.	*Using the Broker and RPC User ID/Password*
<exx-password>	The password for secured access to the broker.	
<exx-password-encryption>	Specifies how the password is encrypted in the configuration file. Encryption is performed automatically when the configuration file is read for the first time. Possible values: base64	plain.	

Parameter	Description	More Information
`<exx-use-security>`	Deprecated. An internal automatic routine makes this parameter obsolete. Possible values: `true` | `false`.	
`<exx-rpc-userID>`	The RPC user ID sent to the RPC server.	*Using the Broker and RPC User ID/Password*
`<exx-rpc-password>`	The RPC password sent to the RPC server.	
`<exx-natural-security>`	Enable to send the RPC user ID/password pair. Possible values: `true` | `false`	
`<exx-natural-library>`	By default the library name sent to the RPC server is retrieved from the IDL file (see `library-definition` under *Software AG IDL Grammar* in the IDL Editor documentation). The library name can be overwritten. This is useful if communicating with a Natural RPC server.	

Example

```
<?xml version="1.0" encoding="utf-8" ?>
<serviceGroup>
  <service name="*">
    <exx-use-security>true</exx-use-security>
  </service>
  <service name="AService">
    <exx-brokerID>myhost:1977</exx-brokerID>
    <exx-service>RPC/ASERVICE/CALLNAT</exx-service>
    <exx-userID>test</exx-userID>
  </service>
</serviceGroup>
```

Using the Broker and RPC User ID/Password

EntireX supports two user ID/password pairs: a broker user ID/password pair and an (optional) RPC user ID/password pair sent from RPC clients to RPC servers. With EntireX Security, the broker user ID/password pair can be checked for authentication and authorization.

The RPC user ID/password pair is designed to be used by the receiving RPC server. This component's configuration determines whether the pair is considered or not. Useful scenarios are:

- Credentials for Natural Security

- Impersonation in the respective RPC Server documentation

- Web Service Transport Security with the RPC Server for XML/SOAP, see *XML Mapping Files*

- Service execution with client credentials for EntireX Adapter Listeners, see *Configuring Listeners*

- etc.

Sending the RPC user ID/password pair needs to be explicitly enabled by the RPC client. If it is enabled but no RPC user ID/password pair is provided, the broker user ID/password pair is inherited to the RPC user ID/password pair.

With the parameter `<exx-natural-security>` (see *Using an External Configuration File*) sending the RPC user ID/password pair is enabled for the RPC Server for XML/SOAP. If you do so, we strongly recommend using SSL/TLS. See *SSL/TLS, HTTP(S), and Certificates with EntireX*.

≫ **To use the broker and RPC user ID/password**

1   Specify a broker user ID with parameter `<exx-userID>`.

2   Optional. Specify broker password with parameter `<exx-password>`.

3   Enable to send the RPC user ID/password pair with parameter `<exx-use-natural-security>` set to `true`.

4   If different user IDs and/or passwords are used for broker and RPC, use the parameters `<exx-rpc-userID>` and `<exx-rpc-password>` to provide a different RPC user ID/password pair.

5   By default the library name sent to the RPC server is retrieved from the IDL file (see `library-definition` under *Software AG IDL Grammar* in the IDL Editor documentation). The library name can be overwritten. This is useful if communicating with a Natural RPC server. Specify a library in parameter `<exx-natural-library>`.

If you need to log on to a Natural library without setting security credentials, refer to *Logon to Natural Library*.

Logon to Natural Library

By default the library name sent to the RPC server is retrieved from the IDL file (see `library-definition` under *Software AG IDL Grammar* in the IDL Editor documentation). The library name can be overwritten. This is useful if communicating with a Natural RPC server.

≫ **To overwrite the library name**

1   Specify the library in the parameter `<exx-natural-library>`.

2   Set the parameter `<exx-natural-security>` to "true".

If you need to set security credentials, see *Using the Broker and RPC User ID/Password*.

Installing Web Services Stack in a Web Server

Deploying Web Services Stack Runtime to Tomcat

If you want to deploy Web Services Stack Web application and EntireX XML runtime to your own Apache Tomcat installation, perform the following steps:

≫ **To deploy to your own Apache Tomcat installation**

1 Stop Apache Tomcat.

2 Make sure that the attribute `unpackWars` is set to "true" in the server.xml of your Apache Tomcat installation.

3 Copy *wsstack.war* to the *webapps* directory of your Apache Tomcat installation.

4 Start Apache Tomcat. The content of *wsstack.war* will be expanded into a directory *wsstack* under the *webapps* directory of the Apache Tomcat installation.

5 Stop Apache Tomcat.

6 Copy *entirex.jar* from the *classes* directory of your EntireX installation to the *WEB-INF/lib* directory of the expanded Web Services Stack Web application.

7 Start Apache Tomcat.

Now you can deploy generated EntireX Web services (.aar) from the Designer.

Using Internationalization with the Listener for XML/SOAP

- RPC Call (Broker Request)
- Outgoing XML Document (Response)

RPC Call (Broker Request)

The encoding for the RPC call (broker request) depends on the setting of following parameters in the given order:

1. HTTP header/parameter `exx-use-codepage` travelling with the XML document incoming to the Listener for XML/SOAP. Valid values:

`true`	The encoding is retrieved from the incoming XML document.
`false`	The encoding of Java system property `file.encoding`.
`<character encoding>`	the given character encoding is used.

See also *The HTTP Interface* in the XML/SOAP Wrapper documentation.

2. Parameter `Use Codepage`, see Generating a Web Service, Step 4, check **Set connection and security parameters in mapping file**.

3. Parameter `Use Codepage`, see EntireX Settings Page under *Web Services Stack Configuration Editor.*

Which encodings are valid depends on the version of your JVM. For a list of valid encodings, see *Supported Encodings* in your Java documentation. The codepage derived after the broker's locale string mapping process must be the same as the one used to encode the data and it must be supported by the broker. See *Locale String Mapping*.

Enable character conversion in the broker by setting the service-specific attribute `CONVERSION` to `"SAGTRPC"`. See also *Configuring ICU Conversion* under *Configuring Broker for Internationalization* in the platform-specific Administration documentation. More information can be found under *Internationalization with EntireX*.

Outgoing XML Document (Response)

By default, the incoming XML encoding is used for the outgoing XML document (response). This can be disabled and an encoding defined in the XML mapping file is used. See *Defining the XML Encoding* in the XML Mapping Editor documentation.