Writing Web Services Applications

This chapter describes the typical steps required to create and register a Web service with the *EntireX Workbench* and how it is made available to Web service client applications. The following topics are covered:

- Generation of EntireX Web Services
- Deploying EntireX Web Services
- Deploying Web Services Stack Runtime
- Developing Web Service Client Applications
- Testing EntireX Web Services
- Removing Web Services

See also Web Services Stack documentation, also available under *http://documentation.softwareag.com* > *webMethods Product Line* > *Shared Services*.

Generation of EntireX Web Services

The EntireX plug-ins for Eclipse provide support to expose EntireX services as Web services. We assume a typical scenario where an existing (legacy) server application has been "wrapped" with EntireX technology and is accessible to clients via the EntireX RPC protocol. The interface of the service is described by an IDL file (see *Software AG IDL File*). If there is a related client-side mapping file (Natural | COBOL), this is also used (internally). Using the EntireX Web service plug-ins for Eclipse, application developers can generate from an IDL file (and optionally a server mapping file), for example, *example.idl* (and *example.cvm*), the following:

- a SOAP mapping (*example.xmm*)
- a WSDL description (*example.wsdl*)
- a service archive for the Web Services Stack (*example.aar*)

These files are used by the EntireX XML/SOAP Runtime to implement a Web service interface for the given service.

From the context menu of your IDL file in the Package Explorer view, choose **Web Service > Generate Web Service...**.

Deploying EntireX Web Services

- Requirements
- Deploying the Web Service
- Undeploying the Web Service

Requirements

The following resources are required to deploy and run an EntireX Web service:

- An application server where the Web Services Stack Web runtime is installed (wsstack.war). The Web Services Stack is typically accessible at the URL *http://<host-name>:<port-number>/wsstack/*. The default port number is 10010. This can be changed during installation. In the case of deployment in custom application servers, the port is configured by the corresponding server administration tools. For more details see the Web Services Stack documentation, also available under *http://documentation.softwareag.com* > *webMethods Product Line* > *Shared Services*.
- The EntireX XML/SOAP Runtime (packaged as Java library *entirex.jar*). File *entirex.jar* must be located in the *WEB-INF\lib* folder of the Web Services Stack Web application.
- The Eclipse plug-ins of the Web Services Stack must be installed.
- EntireX Broker and RPC server hosting the server implementation are up and running.

Deploying the Web Service

Deploying an EntireX service means sending a service archive (extension .aar) to a running Web Services Stack Web application. The Web Services Stack Web application stores the service archive in the *WEB-INF/services* folder of the Web Services Stack Web application.

From the context menu of the generated service archive, choose **Software AG Common Web Services Stack >Deploy Web Service Package**. Now a wizard starts where you can select hostname, port number, and a servlet address of the Web Services Stack Deployment Servlet. You also need to authenticate with your credentials (user ID and password). On **Finish**, the service archive is sent to the selected deployment connection point.

Note:
For more information, see *Deploying and Undeploying Web Service Archives* in section *Software AG Designer Plug-in* of the *Web Services Stack* documentation.

You can verify the deployment of your service with context-menu item **Software AG Common Web Services Stack >View Web Services Stack...** or **Software AG Common Web Services Stack >View Web Service**.

Note:
After the installation of EntireX, only the default deployment connections are available in a new workspace. The default values are:
Hostname: localhost
Port number: 10010 (this number can be changed during installation)
Servlet address: /wsstack/sagdeployer

Choose **Windows** > **Preferences** > **Software AG** > **Web Services Stack** > **Deployment** to add, edit, or remove deployment connections.

Undeploying the Web Service

Undeploying an EntireX Web service means informing a running Web Services Stack Web application to remove a deployed service. The Web Services Stack Web application removes the corresponding service archive from the *WEB-INF/services* folder of the Web Services Stack Web application. You can verify the undeployment with the help of a browser. The undeployed service should disappear from the list of the deployed services (e.g. *http://localhost:10010/wsstack/services/listServices*).

Deploying Web Services Stack Runtime

This section covers the following topics:

- Deploying Web Services Stack Runtime to Tomcat
- Deploying Web Services Stack Runtime to WebSphere 8.5
- Deploying Web Services Stack Runtime to WebLogic 12c

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 EntireX Workbench.

Deploying Web Services Stack Runtime to WebSphere 8.5

If you want to deploy the Web Services Stack Web application to WebSphere 8.5, perform the following steps:

To deploy to WebSphere 8.5

1. Copy *Software AG_directory/WS-Stack/webapp/wsstack.war* to a temporary location.
2. Unpack the WAR file.
3. Copy all MAR files from *WEB-INF/modules* to *WEB-INF/lib* and change their extensions to JAR.

   Important:
   There might be an issue with mapping of MAR files when using Microsoft Office. When you have Microsoft Office installed, you cannot rename those files using Windows Explorer. In this case, use the *ren* command prompt command. For example, `<TEMP_Directory>\WEB-INF\modules>copy addressing-1.41.mar addressing-1.41.jar` copies the MAR file and changes its extension from MAR to JAR.
4. Copy *entirex.jar* to folder*../wsstack/WEB-INF/lib*.
5. Recreate the WAR file. You can use WinZip or any other application with support for ZIP files.
6. Log on to the Administrative Console and navigate to **Applications** > **New Application** > **New Enterprise Application**.
7. Enter the location of the *wsstack.war* file or browse for it using the **Browse** button. Then click **Next**.
8. Select the **Fast Path - Prompt only when additional information is required** radio button and then click **Next**.
9. Click **Next** and leave the default values for the options in the **Step 1 Select installation options**, **Step 2 Map modules for servers**, and **Step 3 Map virtual hosts for Web modules** screens.
10. Click **Next** to go to the **Step 4 Map context roots for Web modules** screen, and type in "wsstack" in the **Context root** field.
11. Click **Save** to save the changes to the master configuration.
12. Navigate to **Applications** > **Application Types** > **WebSphere Enterprise Applications**.
13. Click **wsstack_war** to open the configuration dialog.
14. In the configuration dialog, click the **Class loading and update detection** link.
15. For **Class loader order**, select **Classes loaded with local class loader first (parent last)** radio button.
16. For **WAR class loader policy**, select **Single class loader for application** radio button.
17. Navigate to **Applications** > **Application Types** > **WebSphere Enterprise Applications**.
18. Start the Web Services Stack Web application.

Deploying Web Services Stack Runtime to WebLogic 12c

To deploy *wsstack.war* as Web application for WL server

1. Explode *wsstack.war* in a directory named `wsstack`.
2. Add a file named *weblogic.xml* to the folder *../wsstack/WEB-INF*. *weblogic.xml* has the following content:

   ```
   <weblogic-web-app>
        <container-descriptor>
    <prefer-web-inf-classes>true</prefer-web-inf-classes>
        </container-descriptor>
   </weblogic-web-app>
   ```

3. Copy *entirex.jar* to folder*../wsstack/WEB-INF/lib*.
4. Open the weblogic console (e.g. *http://localhost:7001/console*.
5. Select **Deployments**.
6. Choose **Install**.
7. Select **wsstack(open directory)** as current location and choose **next**.
8. Select **Install this deployment as an application** and choose **next**.
9. Select **DD only** for **security** and **I will make the deployment accessible from the following location** for **Source accessibility**.
10. Enter the path to the *wsstack* folder and choose **next**.
11. Choose **Finish**.

To deploy `example.aar` in wsstack Web application

- As a prerequisite, `Basic Authentication` for your WL server has to be disabled. There are two ways to do this:
  - Edit *config.xml* of your WL server and add `<enforce-valid-basic-auth-credentials>false</enforce-valid-basic-auth-credentials>` under `<security-configuration>`.
  - Use the WebLogic scripting tool to modify the `enforce-valid-basic-auth-credentials` attribute.

    ```
    connect('weblogic','weblogic','t3://localhost:7001')
    edit()
    startEdit()
    ls()
    cd('SecurityConfiguration')
    ls()
    cd('base_domain')
    ls()
    set('EnforceValidBasicAuthCredentials','false')
    ```

```
ls()
save()
activate()
print 'Now Restart Your Server...'
```

Developing Web Service Client Applications

Once the EntireX Web service is up and running and its WSDL is accessible (via HTTP), Web service client applications can be developed. For example, in XMLSpy, in the menu bar choose **SOAP > Create new SOAP request** and enter the URL of the service WSDL (e.g. *http://localhost:10010/wsstack/services/example?wsdl*). XMLSpy reads the WSDL and displays a list of methods. Select one, e.g. CALC, and fill in parameter values in the displayed SOAP message template. Finally select **SOAP > Send request to server** to execute the Web service call.

Testing EntireX Web Services

From the context menu of the generated service archive, choose **Test EntireX Web Service**. This starts the XML Tester.

Note:
The XML Tester must already be configured for the generated and deployed service.

If the archive contains more than one XMM file describing the service, select the XMM file you want. Follow the instructions for the XML Tester to create a sample document.

See *XML Tester*.

Removing Web Services

When a Web service is removed from the project, using **Web Services Stack > Remove Web Service**, the following artifacts are additionally deleted depending on the source of the generated Web service.

Generated from	**Additionally Deleted Artifacts**
Natural/COBOL	● IDL file ● XMM file ● Server mapping file (if applicable, see *Client-side Mapping* for Natural | COBOL) ● WSDL file
IDL File	● XMM file ● WSDL file
XMM File	● WSDL file (if applicable)