

Web Services Stack Help

Version 10.1

October 2017

This document applies to Web Services Stack Version 10.1 and to all subsequent releases.

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

Copyright © 2010-2017 Software AG, Darmstadt, Germany and/or Software AG USA Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at <http://softwareag.com/licenses>.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <http://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at <http://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

Table of Contents

About this Guide.....	5
Document Conventions.....	5
Online Information.....	6
Overview.....	7
About the Web Services Stack Designer Plug-in.....	8
Installing the Web Services Stack Plug-in on an Independent Eclipse Installation.....	8
Verifying the Installed Web Services Stack Application.....	9
Configuring Web Services Stack Preferences.....	11
Configuring XML View Preferences.....	12
Configuring Message Receivers, User Policies, and Password Callback Handlers.....	12
Configuring Message Receivers.....	12
Configuring User Policies.....	13
Configuring Password Callback Handlers.....	13
Configuring Deployment Locations.....	14
Configuring SSL Settings.....	15
Managing Web Service Packages.....	17
Creating Web Service Packages.....	18
Using the Web Services Stack Package Builder.....	18
Deleting Web Service Packages.....	19
Configuring Web Service Packages.....	19
Adding Supplementary Source Files to Web Service Packages.....	20
Updating the Configuration of Web Services within Web Service Packages.....	20
Configuring Web Services Operations within Web Service Packages.....	21
Configuring Advanced Policy Settings.....	23
Overview.....	24
Enabling Web Service Addressing.....	24
Enabling Web Service Reliable Messaging.....	24
Configuring Web Service Security.....	24
Saving Policy Configurations.....	27
Deploying and Undeploying Web Service Packages.....	29
Overview.....	30
Deploying Web Service Packages.....	30
Displaying Deployed Web Service Packages.....	31
Undeploying Web Service Packages.....	31
Publishing and Unpublishing Web Service Packages to CentraSite.....	33
Overview.....	34
Publishing a Web Service Package.....	34

Unpublishing a Web Service Package..... 36

About this Guide

This document contains information about using the Web Services Stack plug-in available within Software AG Designer.

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
Narrowfont	Identifies storage locations for services on webMethods Integration Server, using the convention <i>folder.subfolder:service</i> .
UPPERCASE	Identifies keyboard keys. Keys you must press simultaneously are joined with a plus sign (+).
<i>Italic</i>	Identifies variables for which you must supply values specific to your own situation or environment. Identifies new terms the first time they occur in the text.
Monospace font	Identifies text you must type or messages displayed by the system.
{ }	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

Software AG Documentation Website

You can find documentation on the Software AG Documentation website at <http://documentation.softwareag.com>. The site requires Empower credentials. If you do not have Empower credentials, you must use the TECHcommunity website.

Software AG Empower Product Support Website

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](#).

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.

1 Overview

- About the Web Services Stack Designer Plug-in 8
- Installing the Web Services Stack Plug-in on an Independent Eclipse Installation 8
- Verifying the Installed Web Services Stack Application 9

About the Web Services Stack Designer Plug-in

The Web Services StackDesigner plug-in enables you to create, configure, deploy, and publish Web services. You can expose Web services created with pure Java classes to implement business logic, or create Web service packages containing multiple Java classes essential for the completion of a particular business case.

You create Web services out of Java classes and WSDL files contained in your Java projects in Designer. The Java classes and WSDL files are added to Web service packages together with supplementary source and meta-information files.

When you create Web services, you can design them by providing individual configuration settings. You can modify their behavior at run time and facilitate the correct invocation of the functionality that they expose. You can configure the Web services by providing advanced design settings, such as Web service addressing, security, and transactional behavior.

The Web Services StackDesigner plug-in enables you to deploy your Web services and expose their functionality for a broader usage. Depending on your environment, you can deploy them to the default Web Services Stack servlet container and run them locally, or you can deploy them to a fully functional application server and expose their functionality to other Web service clients.

You can use the Web Services StackDesigner plug-in to publish Web services to CentraSite. To do this, you must have CentraSite Registry Repository and the CentraSite plug-in installed on your system.

Installing the Web Services Stack Plug-in on an Independent Eclipse Installation

By default, the Web Services Stack plug-in is installed with Software AG Designer. However, you can also use your own independent Eclipse installation with the Web Services Stack plug-in.

As a prerequisite, you must use Software AG Installer to install the Web Services Stack plug-in. The installed plug-in package is located in the *Software AG_directory/eclipse/updates* directory. You must then update your independent Eclipse installation with the respective Web Services Stack package.

To install the Web Services Stack plug-in on an independent Eclipse installation

1. Start Eclipse.
2. Navigate to **Help > Install New Software**.
The **Available Software** screen appears.

3. For the `com.softwareag.common.zip` archive, complete the following steps:
 - a. Click **Add**.
 - b. Click **Archive**.
 - c. Browse to the *Software AG_directory* /`eclipse/updates` directory and open the archive.
 - d. Click **OK**.
 - e. Select the components you want to install and click **Next**.
4. For the `eclipse.wss.<version_number>.UpdatePackage.0000.zip` archive, complete the following steps:
 - a. Click **Add**.
 - b. Click **Archive**.
 - c. Browse to the *Software AG_directory* /`eclipse/updates` directory and open the archive.
 - d. Click **OK**.
 - e. Select the components you want to install and click **Next**.

Note: The version number of the particular package depends on the version of the current installation. For example, if you want to update a Web Services Stack 9.0 SP1 installation, the version number of the update package is 901, and the file name is `eclipse.wss.901.UpdatePackage.0000.zip`.

5. Follow the instructions.

Verifying the Installed Web Services Stack Application

You can verify that your Web Services Stack application is installed.

To verify the Web Services Stack installation

1. In Designer, open the Project Explorer view.
2. Right-click a Java project and select **Web Services Stack > View Web Services Stack**.

The **Web Services Stack Locations** window appears, displaying the configuration information provided in the **Windows > Preferences > Software AG > Web Services Stack > Deployment** window.
3. Select an entry from the table and click **OK**.

Designer opens the selected Web Services Stack configuration in an embedded web browser.

4. Click the **Validate** link on the Web Services Stack web application Welcome screen to ensure that all the required libraries are in place.

2 Configuring Web Services Stack Preferences

■ Configuring XML View Preferences	12
■ Configuring Message Receivers, User Policies, and Password Callback Handlers	12
■ Configuring Deployment Locations	14
■ Configuring SSL Settings	15

Configuring XML View Preferences

The Web Services StackDesigner plug-in displays the content of XML documents. If you want to customize the way the content is displayed, you can modify several visualization parameters.

To configure the Web Services Stack XML view preferences

1. In Designer: **Windows > Preferences**.
2. In the preferences navigation tree, select **Software AG > Web Services Stack**.
3. In the Web Services Stack window, set values for the following settings:

Property	Description
Line Width	Specifies the width of the line in symbols. The default value is 132.
Indentation size	Specifies the indentation of the XML elements on the line in symbols. The default value is 2.

4. Click **Apply**, and then **OK** to confirm.

Configuring Message Receivers, User Policies, and Password Callback Handlers

To ensure the communication between applications, services and operations within your web services use message receivers, user policies, and password callback handlers. You can use the default ones, provided with your Software AG installation, or you can configure your custom message receivers, user policies, and password callback handlers.

Configuring Message Receivers

Message receivers are agents that exchange messages in accordance with specific message exchange patterns. You can configure your custom message receivers using the procedure below.

To configure Web Services Stack message receivers

1. In Designer: **Windows > Preferences**.
2. In the preferences navigation tree, select **Software AG > Web Services Stack > Configuration Editor**.

3. In the **Message Receivers** tab, configure the Web Services Stack message receivers.
 - a. In the **Message Exchange Pattern** table, click **Add**, provide a name for the Message Exchange Pattern, then click **Add** to confirm.
 - b. In the **Message Receiver** table, click **Add**, provide a name for the Message Receiver, then click **Add** to confirm.
 - c. To edit or remove an item from the tables, select an entry and click the **Edit** or **Remove** button.

The message receivers that you configure appear in the advanced policy settings of the Web service packages. For more information about Web services packages, see [Managing Web Service Packages](#).

4. Click **Apply**, and then **OK** to confirm your Web Services Stack configuration settings.

Configuring User Policies

User policies are XML files that contain pre-configured user settings and policies, and are previously exported to the file system. You can configure your custom user policies using the procedure below.

To configure Web Services Stack user policies

1. In Designer: **Windows > Preferences**.
2. In the preferences navigation tree, select **Software AG > Web Services Stack > Configuration Editor**.
3. In the **User Policies** tab, configure the Web Services Stack user policies.
 - a. In the **User Policies** tab, click **Add**.
 - b. Open the policies XML file, then click **Add** and **OK** to confirm.
 - c. To edit or remove an item from the tables, select an entry and click the **Edit** or **Remove** button.

The user policies that you configure appear in the advanced policy settings of all Web service packages. For more information about advanced policy settings, see [Configuring Advanced Policy Settings](#).

4. Click **Apply**, and then **OK** to confirm.

Configuring Password Callback Handlers

Password callback handlers provide or validate credentials for a specific identifier. You can create your custom password callback handlers using the procedure below.

To configure Web Services Stack password callback handlers

1. In Designer: **Windows > Preferences**.

2. In the preferences navigation tree, select **Software AG > Web Services Stack > Configuration Editor**.
3. In the **Callback Handlers** tab, configure the Web Services Stack callback handlers.
 - a. In the **Password Callback Handler** table, click **Add**, provide a name for the password callback handler, then click **Add** to confirm.
 - b. In the **Policy Validator** table, click **Add**, provide a name for the policy validator, then click **Add** to confirm.
 - c. To edit or remove an item from the tables, select an entry and click the **Edit** or **Remove** button.

The callback handlers that you configure appear in the advanced policy settings of the Web service packages. For more information about Web services packages, see [Configuring Web Services Stack Preferences](#).
4. Click **Apply**, and then **OK** to confirm.

Configuring Deployment Locations

After you create and configure a Web Services Stack package, you can deploy it on the Web Services Stack instance, which is deployed either on Software AG Runtime, or on a standard Java servlet container. The deployed package can then be exposed to consumers. Before you deploy a Web service package, you must define the deployment locations.

To configure the Web Services Stack deployment locations

1. In Designer: **Windows > Preferences**.
2. In the preferences navigation tree, select **Software AG > Web Services Stack > Deployment**.
3. In the **Deployment** window, click **Add**, provide values for the properties in the table below, and click **OK** to confirm.

Property	Description
Name	A name for the connection.
Protocol	A protocol for the connection. Possible options are: <code>http</code> (default) and <code>https</code> .
Host	A host name. Default value is <code>localhost</code> .
Port	A port number for the connection. The default value is <code>10010</code> .

Property	Description
Servlet	Location of the deploy servlet. The default value is <code>/wsstack/sagdeployer</code> .
User	A user name.
Password	A password.
Store password in secure storage	

4. Click **Apply**, and then **OK** to confirm.

Configuring SSL Settings

You can configure your Web Services StackDesigner plug-in to use Secure Sockets Layer (SSL) to encrypt the information exchange over a secure protocol.

To configure Web Services Stack SSL settings

1. In Designer: **Windows > Preferences**.
2. In the preferences navigation tree, select **Software AG > Web Services Stack > SSL**.
3. In the **SSL** window, browse to select the truststore file location, and enter a value for the truststore password.
4. To specify a keystore file that differs from the truststore, clear the **Use truststore settings for keystore** check box. Browse to select the keystore file location, and enter a value for the keystore password.
5. Click **Apply**, and then **OK** to confirm.

3 Managing Web Service Packages

- Creating Web Service Packages 18
- Deleting Web Service Packages 19
- Configuring Web Service Packages 19

Creating Web Service Packages

You create Web service packages out of Java classes and WSDL files contained in your Java projects in Designer. A single Java project can contain multiple Web service packages which differ in configuration and exposed functionality. Later you can deploy the created Web service package to an application server to expose its Web services and make their functionality available to other applications.

To create a web service package

1. In Designer, open the **Project Explorer** view.
2. In the **Project Explorer**, right-click the Java project, where you want to create a Web service package.
3. On the context menu, select **Web Services Stack > Add Web Services Package**.
4. In the **Web Services Packaging Wizard**, specify a container and a unique name for the Web service package, and click **Next**.

By default, the container is the Java project.

5. Select and open the source file which you want to expose as a Web service, and click **Next**.

You can select a WSDL file or a Java class file as a source file on the file system.

Note: The source file you expose must be in the same Java project where you create the Web service. You cannot expose external files as Web services in the current Java project.

6. Optionally, select supplementary meta-information files, and click **Next**.

The files you select in this step are included in the META-INF directory of the Web service package.

7. Optionally, select supplementary source files, and click **Next**.

The files you select in this step are included in the root directory of the Web service package.

8. Click **Finish**.

Using the Web Services Stack Package Builder

The Web Services StackDesigner plug-in has its own Web Services Stack Package builder, similar to the Java builders. The builder stores some metadata information in a hidden folder, called .wstack. Every time a package is deleted or any of the files within the Java project are modified, the builder recreates the package. For example, if a Java class that is part of a package is modified and saved, the Java builder recreates the corresponding class file, and the Web Services Stack Package builder recreates the

package with the new class file. To deactivate the builder for the particular project, right-click and select **Properties > Builders**.

If a file, which is part of a Web service package, is deleted, renamed, or moved, the Web Services Stack Package builder asks you if you want to keep the file in the package.

- If you decide to keep it, the file is copied to the corresponding temporary WSS directory, and you can rebuild the package.
- If you decide not to keep it, the file is removed from the package.

Note: If you keep the file in the temporary directory, and then you add a new file with the same name to the project, the build process allows you to add the new or the restored file to the Web service package. The old file is kept in the package and it is removed from the temporary directory.

Deleting Web Service Packages

Using the Web Services StackDesigner plug-in, you can delete Web service packages.

To delete a Web service package

1. In Designer, open the **Project Explorer** view.
2. In the **Project Explorer**, do one of the following:
 - Right-click the Web service package you want to delete, and select **Web Services Stack > Remove Web Services Package**.

Caution: Make sure you select the correct Web service package for deletion. Designer does not ask you to confirm your choice and you cannot undo the action.

- Right-click a Java project, and then select **Web Services Stack > Remove Web Services Package**. Select the Web service packages you want to delete, and click **OK**.

Configuring Web Service Packages

Using the Web Services StackDesigner plug-in, you can configure the Web service packages, as well as the services and the service operations within them.

The configuration settings are grouped in the Web service package editor under the following tabs:

- **Archive** - configures the Web service package.
- **Services** - configures the Web services within the Web service package.
- **Operations** - configures the Web service operations within the Web service package.

- **Services.xml** - displays a read-only textual view of the Web service.

Note: You can also display a tree structure of the Web service package using the **Outline** view in Designer and navigate through the different tabs by clicking the respective element in the tree. To open the Outline view, select **Window > View > Outline**.

Adding Supplementary Source Files to Web Service Packages

You can add supplementary source files and Java classes to existing Web service packages using the **Archive** tab in the Web service package editor.

To add supplementary source files to a Web service package

1. In Designer, open the **Project Explorer** view.
2. Double-click to open the Web service package you want to configure.
By default, the Web service package editor opens the **Archive** tab.
3. To add a supplementary source file to the META-INF directory:
 - a. In the **Archive Structure** area, click **Add**.
 - b. Select and open the source file you want to add as a META-INF file, and click **OK**.
4. To add a supplementary source file and expose it as a Web service:
 - a. In the **Web Service Files** area, click **Add**.
 - b. Select and open the source file you want to expose as a Web service, and click **OK**.

You can select a WSDL file or a Java class file as a source file on the file system.

Note: The source file you expose must be in the same Java project where you create the Web service. You cannot expose external files as Web services in the current Java project.

5. Save your changes.

Updating the Configuration of Web Services within Web Service Packages

You can update and provide further configuration settings for Web services already included in Web service packages using the **Services** tab in the Web service package editor.

To update the configuration of Web services within a Web service package

1. In Designer, open the **Project Explorer** view.
2. Double-click to open the Web service package you want to configure.

3. Click the **Services** tab.
4. In the **Service** drop-down list, select the Web service you want to configure.
5. In the **Description** area, add a description for the Web service, and click **Set** to confirm.
6. In the **Properties** area:
 - Click the **Add** button to add a new property. Select a name, set a value, then click **OK**.
 - Select an existing property and click **Edit** to modify it. Set a new name and value, then click **OK**.
 - Select an existing property and click **Remove** to delete it.
7. In the **Message Receivers** area:
 - Click the **Add** button to add a new message receiver. Select a name, set a value, then click **OK**.
 - Select an existing message receiver and click **Edit** to modify it. Set a new name and value, then click **OK**.
 - Select an existing message receiver and click **Remove** to delete it.
8. In the **Transport** area, select the transport you want to use with the Web service. You can select between the following transport types: **HTTP**, **HTTPS**, **TCP**, and **JMS**.

If you do not select a specific transport type, all transports enabled on the respective Web server are available for the Web service.
9. In the **Modules** area, enable and configure the advanced policy settings for the Web service.

For more information about configuring advanced policy settings, see [Configuring Advanced Policy Settings](#).
10. Save your changes.

Configuring Web Services Operations within Web Service Packages

You can provide additional configuration settings for Web service operations within a Web service package.

To configure Web service operations within a Web service package

1. In Designer, open the **Project Explorer** view.
2. Double-click to open the Web service package you want to configure.
3. Click the **Operations** tab.
4. In the **Service** drop-down list, select the Web service you want to configure.
5. In the **Operation** drop-down list, select the operation you want to configure.

6. In the **Message Receivers** area:

- Select the class name, then click **Set** to assign a new message receiver.

You can use your own user-defined message receivers. For more information about user-defined message receivers, see [Configuring Message Receivers, User Policies, and Password Callback Handlers](#).

- Click the **Clear** button to remove an assigned message receiver.

7. In the **Parameters** area:

- Click the **Add** button to add a new parameter. Select a name, set a value, then click **OK**.
- Select an existing parameter and click **Edit** to modify it. Set a new name and value, then click **OK**.
- Select an existing parameter and click **Remove** to delete it.

8. In the **Modules** area, enable and configure the advanced policy settings for the Web service.

For more information about configuring advanced policy settings, see [Configuring Advanced Policy Settings](#).

9. Save your changes.

Note: In a POJO Web service, all public methods are exposed as operations, even if they are not declared in the services.xml file. To exclude an operation, edit manually the services.xml file, and add the following excerpt:

```
<excludeOperations>
  <operation>[operation_name]</operation>
</excludeOperations>
```

4 Configuring Advanced Policy Settings

■ Overview	24
■ Enabling Web Service Addressing	24
■ Enabling Web Service Reliable Messaging	24
■ Configuring Web Service Security	24
■ Saving Policy Configurations	27

Overview

The Web Services Stack tool enables you to configure Web services and Web service operations advanced policy settings, such as: WS-Addressing, WS-ReliableMessaging, and WS-Security. You can also save your policy configuration as a policy XML file, and reuse it later.

Enabling Web Service Addressing

To enable WS-Addressing

1. In Designer, double-click a Web service package to open it.
2. Select the **Services** or **Operations** tab, depending on which one you want to configure.
3. Select **Enable WS-Addressing**.
4. Save your changes.

Enabling Web Service Reliable Messaging

To enable WS-ReliableMessaging

1. In Designer, double-click a Web service package to open it.
2. Select the **Services** or **Operations** tab, depending on which one you want to configure.
3. Select **Enable WS-ReliableMessaging**.
4. Set the inactivity timeout period in milliseconds. The default value is 600000.
5. Save your changes.

Configuring Web Service Security

To configure WS-Security

1. In Designer, double-click a Web service package to open it.
2. Select the **Services** or **Operations** tab, depending on which one you want to configure.
3. Select **Enable WS-Security**

The **Security** opens and the **Enable WS-Addressing** option, the **HTTPS** transport, **Transport-level Security with SSL** and **Username Token Assertion** are automatically enabled.

4. In the **Security** area, you can select one of the four Security Bindings:

- **No Binding** - when selected, you can disable the **Enable WS-Addressing** option.
- **Transport-level Security with SSL**
- **Message-level Security with Symmetric Binding**
- **Message-level Security with Asymmetric Binding**

Depending on the security binding you selected, continue with the relevant options below.

5. Configure the options in the **Message-level Security Options** area.

Option	Description
Encrypt Body	Enables encryption (EncryptedParts assertion) over the entire body content.
Sign Body	Enables integrity protection (SignedParts assertion) over the entire body content.
Sign entire Headers and Body	Enables signature over the headers and body content (OnlySignEntireHeadersAndBody policy assertion).

6. Configure the options in the **Token Assertions** area.

Option	Description
User Name Token	Enables the UsernameToken assertion to secure the messages sent to the client.
Secure Conversation	Enables the SecureConversationToken assertion Language (WS-SecureConversation). WS-SecureConversation provides means for creating and sharing security contexts rather than relying only on the message authentication model.

7. Other Security Options

Option	Description
Include Timestamp	Enables the usage of the IncludeTimestamp policy assertion.
Use Client Certificate	Enables the usage of client certificates to encrypt the transferred data (RequireClientCertificate policy assertion).

Option	Description
--------	-------------

Note: This option is only available when you configure transport-level security with SSL.

8. In case of Message-level Security you can specify the following:

- XPath expressions to encrypt (EncryptedElements or ContentEncryptedElements assertion) and/or sign (SignedElements assertion) parts of the message.
- Headers to be encrypted (EncryptedParts assertion) and/or signed (SignedParts assertion).

Configure the options in the Security Model Configuration (Apache Rampart see <http://axis.apache.org/axis2/java/rampart/rampartconfig-guide.html>) area.

Option	Description
--------	-------------

User	Specifies the user name used. For details see "Client-Side Configuration" in the Web Services Stack documentation.
UsernameToken Validator Class	Callback Class providing the password used to create the UsernameToken for the desired user.
Password Validator Class	Callback Class providing the password to sign the message or to create the UsernameToken (for the details of the predefined callback classes see "Implementation of Password Callback Handlers in the Web Services Stack documentation).
Policy Validator Callback Class	Callback Class providing the password for Custom validators (for the details of the predefined callback classes see "Implementations of Policy Validation Callbacks" in the Web Services Stack documentation).

Optionally, you can add and use your own, user-defined password callback classes, user-defined policy validator callback classes that you added in the preferences settings of Designer. For more information, see [Configuring Web Services Stack Preferences](#).

In the Encryption and Signing areas below you can configure the corresponding keystore and truststore.

Option	Description
Certificate Alias	The user name used for encryption (ENCRYPTION_USER) or for signature (USER_CERTIFICATE_ALIAS).
Keystore	Specifies the path to keystore file for encryption or for signature.
Keystore Password	Specifies the password for the keystore file.
Truststore	Specifies the path to truststore file for encryption or for signature.

9. Save your changes.

For more information see the separate Web Services Stack documentation, also available under <http://documentation.softwareag.com> > webMethods Product Line > Shared Components.

Saving Policy Configurations

To save your policy configuration to a file

1. In Designer, in the **Policies** area, click **Save Policy as**.
2. In the dialog that opens, specify a file name and directory on the file system.
3. Save your changes.

Designer saves your policy configuration and makes it available in the **Available Policies** drop-down menu.

Note: The policy file stored using the configuration editor appears only in the **Available Policies** drop-down menu for this package. Other Web service packages have no access to the policy file. To enable other Web service packages to use the policy file, you must add it to Web Services Stack preferences in Designer. For more information, see [Configuring Web Services Stack Preferences](#).

5 Deploying and Undeploying Web Service Packages

■ Overview	30
■ Deploying Web Service Packages	30
■ Displaying Deployed Web Service Packages	31
■ Undeploying Web Service Packages	31

Overview

When you create and configure a Web service package, you can deploy it on Web Services Stack, which runs on a servlet container.

By default, Web Services Stack runs on a Software AG Runtime instance at port 10010. After installation, the Web Services Stack deployment location is set to `localhost:10010`. You can configure an alternative deployment location using the Web Services StackDesigner plug-in. For more information, see [Configuring Web Services Stack Preferences](#).

Note: We strongly recommend you use only the Software AG Runtime (CTP) with the Web Services StackDesigner plug-in. Software AG does not officially support deployment scenarios for Web Services Stack on other web servers. To install the Software AG Runtime with Web Services Stack, choose **Infrastructure > Shared Platform > Runtime 9.12** in the Installer tree.

Deploying Web Service Packages

You can deploy Web service packages using the Web Services StackDesigner plug-in.

To deploy a Web service package

1. In Designer, open the **Project Explorer** view.
2. To specify the Web service package that you want to deploy, do one of the following:
 - Right-click the Web service package.
 - Right-click the Java project that contains the Web service package.
3. On the context menu, select **Web Services Stack > Deploy Web Services Package**.
The **Software AG Web Services Stack** wizard opens.
4. Provide the following information:

Field	Description
Name	The location to which you deploy your Web service package. The default value is <code>localhost-10010</code> .
URL	The location of the Web Services Stack deploy servlet. The default deploy servlet location is <code>http://<host_name>:<port_number>/wsstack/sagdeployer</code> .

Field	Description
User	The username for the required authentication credentials. The default value is <code>admin</code> .
Password	The password for the required authentication credentials. The default value is <code>manage</code> . For more information about modifying the user name and password, go to Software AG Documentation Website > webMethods Product Suite > webMethods Product Suite [version] > Documentation by Product > Web Services Stack .

- Complete the deployment process by doing one of the following:
 - If you started the wizard on a Web service package, click **Finish**.
 - If you started the wizard on a Java project, click **Next**, select the Web service package you want to deploy, and click **Finish**.

Displaying Deployed Web Service Packages

To display the particular WSDL document, you must deploy the respective Web service package.

To display a deployed Web service package

- In Designer, open the **Project Explorer** view.
- Right-click the Web service package whose WSDL document you want to display.
- On the context menu, select **Web Services Stack > View deployed Web Service**.

Designer opens the WSDL document of the deployed Web service in an embedded web browser.

Undeploying Web Service Packages

To undeploy a Web service package

- In Designer, open the **Project Explorer** view.
- Right-click the Web service package that you want to undeploy.
- On the context menu, select **Web Services Stack > Undeploy Web Services Package**.
- Provide the following information:

Field	Description
Name	The location to which you deploy your Web service package. The default value is <code>localhost-10010</code> .
URL	The location of the Web Services Stack deploy servlet. The default deploy servlet location is <code>http://<host_name>:<port_number>/wsstack/sagdeployer</code> .
User	The username for the required authentication credentials. The default value is <code>admin</code> .
Password	The password for the required authentication credentials. The default value is <code>manage</code> . For more information about modifying the user name and password, go to Software AG Documentation Website > webMethods Product Suite > webMethods Product Suite [version] > Documentation by Product > Web Services Stack .

5. Click **Finish**.

6 Publishing and Unpublishing Web Service Packages to CentraSite

■ Overview	34
■ Publishing a Web Service Package	34
■ Unpublishing a Web Service Package	36

Overview

Software AG Designer enables you to publish previously deployed Web service packages to multiple CentraSite Organizations.

The following statements are valid for the publication process:

- The service you publish is registered in the CentraSite Registry via the WSDL document.
- The WSDL document and the Web service packages are stored in the CentraSite Repository.
- The service object gets an external link to the package.

For more information about deploying services, see [Deploying and Undeploying Web Service Packages](#).

For more information about CentraSite, CentraSite Organizations, and the publication process, see the CentraSite documentation on the [Software AG Documentation website](#).

Publishing a Web Service Package

Using Software AG Designer, you can publish previously deployed Web service packages to CentraSite Organizations. As a prerequisite, you must have the CentraSite Registry Repository and the CentraSiteDesigner plug-in installed on your system.

Before you publish your Web service package to a CentraSite Organization, make sure you have the authentication credentials needed to access that Organization.

To publish a Web service package

1. In Designer, open the **Project Explorer** view.
2. Right-click the Web service package that you want to publish.
3. On the context menu, select **Web Services Stack > Register Web Services Package**.

The **Publish to CentraSite** wizard opens.

4. Provide the following information:

<u>Field Name</u>	<u>Description</u>
Host	The host name on which you can access CentraSite. The default value is localhost.
Port	The port number on which you can access CentraSite. The default value is 53305.

Field Name	Description
Product ID	The ID of the product. The default value is <code>CentraSite</code> .
Repository Path	The path to the repository. The default value is <code>WebServicesStack</code> .
User	The user name for access to a CentraSite Organization.
Password	The password for access to a CentraSite Organization.
Organization	The name for the CentraSite Organization to which you want to publish the Web service. Select a value from the drop-down list.
User	The user name for access to a CentraSite Organization.

- Click **Next**.
- Provide the following information about the deployment location of the Web service:

Field Name	Description
Name	The location to which you deploy your Web service package.
URL	The location of the Web Services Stack deploy servlet. The default deploy servlet location is <code>http://<host_name>:<port_number>/wsstack/sagdeployer</code> .
User	The username for the required authentication credentials. The default value is <code>admin</code> .
Password	The password for the required authentication credentials. The default value is <code>manage</code> . For more information about modifying the user name and password, go to Software AG Documentation Website > webMethods Product Suite > webMethods Product Suite [version] > Documentation by Product > Web Services Stack .

- Click **Finish**.

Unpublishing a Web Service Package

You can unpublish a Web service package from CentraSite Organizations.

To unpublish a Web service package

1. In Designer, open the **Project Explorer** view.
2. Right-click the Web service package that you want to unpublish.
3. On the context menu, select **Web Services Stack > Unregister Web Services Package**.
The **Unregister from CentraSite** wizard opens.
4. Provide the following information:

Field	Description
Product ID	The ID of the product. The default value is <code>CentraSite</code> .
User	The user name for access to a CentraSite Organization.
Password	The password for access to a CentraSite Organization.

5. Click **Finish**.