Web Services Stack Guide

Introduction to WSS

Version 8.0 SP4

March 2010

WEB SERVICES STACK

This document applies to WSS Guide Version 8.0 SP4.

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

Table of Contents

1 Introduction to WSS

This document provides general information on the installed Web Services Stack (WSS).

For details on the installation of Web Services Stack, see the *Installation Guide* on the **Software AG Documentation Web site**. The installation guide provides details on the following information:

- Installation and uninstallation
- Software and hardware support and requirements

Since the general layout of the installed Web Services Stack folders does not differ for Windows and UNIX, the information in this document is relevant for both Windows and UNIX platforms.

The information in this document is organized under the following headings:

2 Overview of Web Services Stack

Overview of Web Services

The use of the term "Web" implies that web services are based on the World Wide Web. Web services, however, are not limited to the Web itself. Web services are a new approach for building, extending, integrating and deploying applications based on XML. This new approach builds applications that can facilitate the communication process over the Internet, as well as the communication process between applications.

Definition of Web Services

Web services are software components that impact application structures. They help the interoperability between platforms that have different programming models. For example, services can hide the details about whether the component is a J2EE-based component or a .NET based component.

According to the definition that is provided by ***http://www.w3.org/TR/ws-gloss/#webservice***, a web service is "a software system designed to support interoperable machine-to-machine interaction over a network". The core of web services is the use of common transports (such as HTTP), data formats (XML, XML Schema), and protocols to create an interoperable way to integrate applications.

Characteristics of Web Services

The information that is published about a web service provides details of what the service is and what it does. This can include information about the transport protocols that can be supported to send a message to the service, the wire format of this message expected by the service, whether and how the message has to be encrypted or signed.

A service is available at a particular endpoint in the network, and it receives and sends messages and exhibits behavior according to its specification.

The functional aspects of a service are specified using WSDL, and the constraints and conditions that are associated with the use of the service are specified through policies and configurations

that you can attach to various parts of the WSDL. These are published so that the users of the service can discover and be given all the information they need to bind to that service.

Summary of Web Services Standards

- HTTP and SMTP for basic network transport services
- XML (Extensible Markup Language) as data format
- Universal Description, Discovery and Integration (UDDI) for web service registries
- The Web Service Description Language (WSDL) for service descriptions
- The Simple Object Access Protocol (SOAP) for XML messaging and RPC
- Web Services Policies

Software AG Web Services Stack (WSS)

Software AG Web Services Stack is a toolkit that provides functionality for execution, configuration and management of web services. It handles the complex process of sending and receiving of web services requests in Software AG products. It allows everyone to do web services while knowing only a few of the details about the web services specifications.

The core part of the Web Services Stack runtime is the SOAP engine, based on Apache Axis 2. Incoming SOAP requests are processed by this SOAP engine. The SOAP request is given to the SOAP runtime and sent back to the client as a SOAP response message. If an error occurs a SOAP fault message is sent back to the client.

The option for generation, configuration and deployment of web services into Software AG Web Services Stack runtime allows the products to "wrap" an existing (legacy) server application and make it accessible to clients. Generally speaking, the products services are exposed as web services and deployed to a running Web Services Stack application.

3 Verifying WSS Installation

To verify Web Services Stack installation

1 Type *http://localhost:49981/wsstack* in the browser.

Note: *49981* is the default CTP port. Check the port number in case of deployment of the Web Services Stack in custom application servers where the port is configured by the corresponding server administration tools.

2 Follow the *Validate* link on the Web Services Stack web application welcome screen to ensure that all the required libraries are in place.

If there are problems with the installation, you see respective red warnings notifying you of the problem.

Note: Change your user credentials after you have logged on to the administration module. Refer to *Changing Logon Credentials* for guidelines on how to do that.

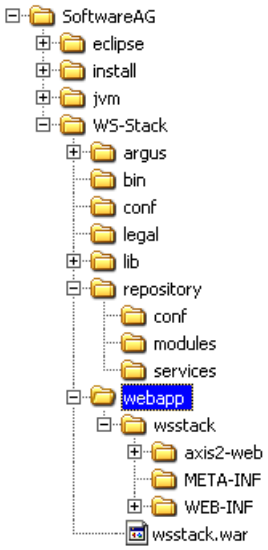
4 Installation Artifacts

The information is organized under the following headings:

Installation Artifacts Under the \SoftwareAG\WS-Stack Folder

This folder contains the Web Services Stack program files.

The WSS installation creates the following directory structure:



The table below lists the contents of each folder.

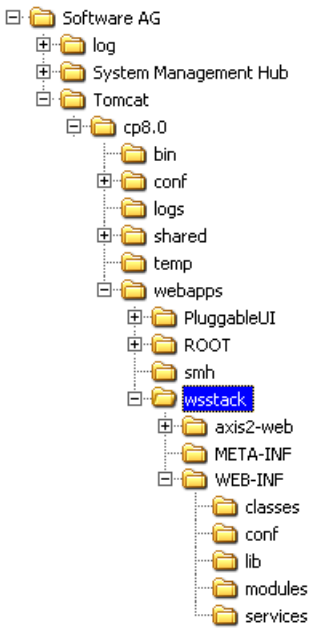
argus	The binary files and other resources needed by Web Services Stack Argus agents
bin	The batch files that facilitate the usage of Web Services Stack command prompt utilities; for example, *axis2.bat*, *axis2server.bat*, *java2wsdl.bat*, *wsdl2java.bat*, etc.
conf	The configuration file needed for the stand-alone server
lib	All the third party libraries and libraries owned by Software AG that are needed for running, building, and developing of web services artifacts (clients and services)
miscellaneous	This directory contains all modules that have experimental or optional status and are not deployed by default. To use the content of the module, copy it manually into the Web Services Stack web application or into the Axis 2 “repository” that is in use by a given client. For example, the Kandula2 module and the needed libraries are stored into the *miscellaneous* folder.
repository	The binary files and text configuration files that are needed on the client side and are spread in separate subdirectories

repository\conf	The configuration files needed to run the WSS client applications
repository\modules	The Web Services Stack modules that provide additional functionality to the engine; for example, security, reliable messaging, etc. The modules are packed into Java archives and bear the MAR file extension (MAR = module archive).
repository\services	The web services either packed into Java archives bearing the AAR file extension (AAR = Axis archive) or in an expanded directory structure **Note:** This is in case one decides to use the installed environment at server side.
webapp	The Web Services Stack web application and a packed WAR file (*wsstack.war*) for an installation in a custom web application container

Installation Artifacts Under the Tomcat Folder

This folder contains the deployed web application in the *webapp* directory of the Tomcat kit (or Apache Tomcat).

The WSS installation creates the following directory structure:



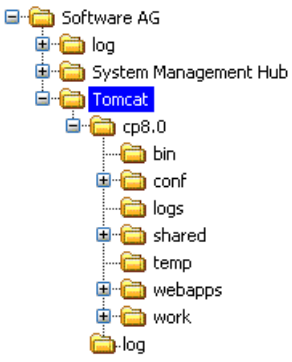
The table below lists the contents of each folder.

axis2-web	The web resources: **.jsp*, **.css*, and **.inc* files
META-INF	The *MANIFEST.MF* file for the web archive
WEB-INF	The core parts of the Web Services Stack engine in separate sub-folders and the *web.xml* (web application configuration descriptor file)
WEB-INF\classes	Any supporting Java classes needed for the proper operation of a deployed web service
WEB-INF\conf	The configuration files
WEB-INF\lib	All Java libraries needed by the Web Services Stack engine
WEB-INF\modules	The modules that provide additional functionality to the engine; for example, security, reliable messaging, etc. The modules are packed in Java archives and bear the MAR file extension (MAR = module archive).
WEB-INF\services	The web services either packed in Java archives bearing the AAR file extension (AAR = axis archive) or in an expanded directory structure

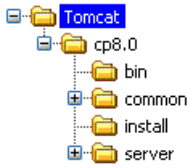
For additional information on deployment, see *Deployment of the Runtime*

Note: On Windows, unlike the Apache Tomcat distribution, the Tomcat kit installs the Tomcat files at two locations. The following screen captures illustrate the installation locations:

- *\Documents and Settings\All Users\Application Data\Software AG\Tomcat*



- *Program Files**Software AG**Tomcat*

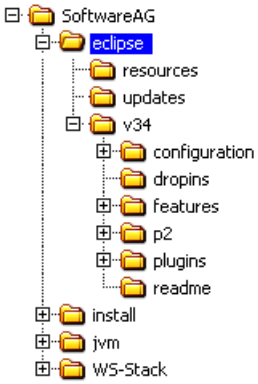


Note: The locations of the configuration files of the Tomcat kit can co-exist with a standard installation of Apache Tomcat server.

Installation Artifacts Under the Eclipse Folder

Software AG Eclipse Package (SEP) creates the *\SoftwareAG\eclipse* folder.

This folder contains the installed Software AG Eclipse plug-ins.



For more details on how to install and use the Web Services Stack Eclipse plug-ins, see *Eclipse Plug-in*.

The version number of the particular update package is a three-digit number which depends on the version of the current installation. You can construct the version number using the version of the release (omitting any periods in the number) and the SP version (using the number of the SP release). For example, if you want to update a Web Services Stack *8.0* SP4 installation, the version number of the update package is *804*, and the file name is: *eclipse.wss.804.UpdatePackage.0000.zip*.

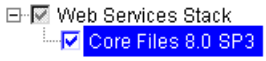
Note: You can use *eclipse.wss.version_number.UpdatePackage.0000.zip* and *com.softwareag.common.zip* under the *\SoftwareAG\eclipse\updates* folder for installation of the Web Services Stack Eclipse plug-ins in another Eclipse distribution. The version number of the particular update package is a three-digit number which depends on the version of the current installation. You can construct the version number using the version of the release (omitting any

periods in the number) and the SP version (using the number of the SP release). For example, if you want to update a Web Services Stack *8.0* SP*4* installation, the version number of the update package is *804*, and the file name is: *eclipse.wss.804.UpdatePackage.0000.zip*.

5 Dependencies of Web Services Stack Components

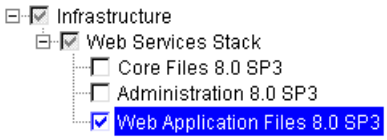
Following is a list of the Web Services Stack components and related dependencies:

- **Web Services Stack under*Designer*.**
  The **Web Services Stack** under *Designer/Web Services Stack* is the module that install the Eclipse Plug-in. It is dependent on *Designer/Shared Plug-ins*.
- **Core Files under*Infrastructure/Web Services Stack***
  No dependencies for installing **Core Files** only.



- **Administration under*Infrastructure/Web Services Stack***
  Administration is dependent on the Web Services Stack program files, the web application, the Java kit, the Tomcat kit, and the System Management Hub.

- **Web Application Files under***Infrastructure/Web Services Stack*
  No dependencies for installing **Web Application Files**.



- **Web Application Deployment under***Infrastructure/Web Services Stack*
  This component is dependent on **Web Application Files**, **Java Development Kit**, and **Tomcat Kit**.

- **The Java Development Kit under***Infrastructure/Web Services Stack*
  This package is required by certain products.
- **The Tomcat Kit under***Infrastructure/Web Services Stack*
  This package is dependent on the **Java Development Kit**.

  The **Tomcat Kit** installs Apache Tomcat Server 5.5 and Software AG-specific security and interface frameworks.
- **System Management Hub under***Infrastructure/Web Services Stack*
  This package is dependent on **Tomcat Kit** and **Java Development Kit**.

6 Upgrading Web Services Stack from a Previous Version

For details on upgrading earlier version of Web Services Stack version 8, see the *webMethods Upgrade Guide* distributed with the webMethods documentation package on the Software AG Installer and posted on the **Software AG Documentation Web site**.

This chapter provides information on the following topics:

Upgrading from Web Services Stack Version 1.2

Web Services Stack ver. 8.x does not upgrade Web Services Stack 1.2. The two versions can co-exist on the same machine.

For example, if you install a product that uses Web Services Stack 8.0.1 on a machine with an installation of Web Services Stack 1.2 (installed previously with another Software AG product), both versions co-exist independently.

Verifying the Upgrade

To verify the success of the upgrade

1. Type *http://localhost:49981/wsstack* in the browser.
2. Follow the *Validate* link on the Web Services Stack web application welcome screen to ensure that all the required libraries are in place.

For details on verifying Web Services Stack Installation, see ***Verifying Web Services Stack Installation***.