

## SDN Community Contribution

(This is not an official SAP document.)

### Disclaimer & Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.

## Applies To:

Web Services, SAP Web Application Server (SAP Web AS)

## Summary

In this article I examine three different tools for testing your SOAP web services. I explain how to use them, highlight their pros and cons, and provide some additional tips. The three tools to examine are: XMLSpy, SAP SOAP client, and the Web Service Navigator (in Web AS).

**By:** Gilles Atlan

**Company:** SAP, SPI

**Date:** 27 Jun 2005

## Table of Contents

Applies To:.....	2
Summary .....	2
Table of Contents .....	2
Tools for Testing Your SOAP Web Services.....	2
SAP Web AS: Web Service Navigator .....	2
Altova XMLSpy .....	3
SAP SOAP Client .....	4
Author Bio.....	4

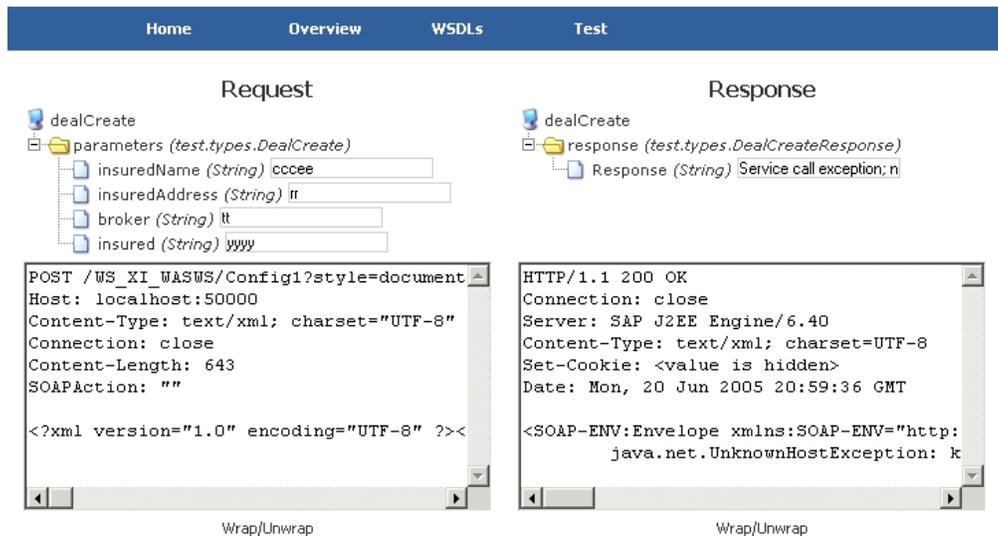
## Tools for Testing Your SOAP Web Services

### SAP Web AS: Web Service Navigator

If you have deployed your web service on Web AS, the easiest way to test your web service is by going to the “WAS Web Service Navigator” at:

`http://<was-hostname>:<port>/wsnavigator` (by default the port 50000)

Once you reach the page, you have a list of web services (WS) that are deployed on Web AS. If your WS does not appear it means that it has not been deployed correctly. The WS name that should appear is the name that you gave it while building it WS in SAP NetWeaver Developer Studio. First, select your WS. You will reach a page where the Web Services Description Language (WSDL) URL is displayed, and in the menu you will have a link called “Test.” Click on Test and you will see your remote function displayed. Click on the remote function that you would like to test, enter the passing parameter, and click Send.



**Figure 1: Sample of a Tested Method on Web AS Web Service**

On the left side you have the SOAP request, and on the right side the SOAP response. In this example (see figure 1), the SOAP WS was not successful because the host that has been called by the WS is unknown (not in the network).

In the menu you can see the “WSDLs” link. From there you can download the WSDL files. SAP provides the WSDL divided in three files; one describing the service, one the port type, and the last one describing the binding. Notice that each WSDL file imports another by using the XML element “import;” however, this element is not supported by all the tools that read WSDL. For example, Adobe Designer cannot load such a WSDL. In that case, you will have to get rid of the import elements by merging the three WSDL files in one.

**Pros:** User friendly; free

**Cons:** Only for WS deployed on Web AS

## Altova XMLSpy

A 30-day trial version of the XMLSpy tool can be downloaded from <http://www.altova.com/download.html>. XMLSpy is very easy to use. Before you test your WS, you can check if your WS WSDL file is valid. Open the WSDL file with XMLSpy and press the F8 key, or select the menu XML -> Validate.

Once your WSDL is valid, go to the SOAP menu and select “Create new SOAP request.” In the new window select your WSDL file and then select the remote function that you want to test. Click OK. XMLSpy will generate the SOAP request from your WSDL file. You can edit it if you wish.

To run the SOAP request, select the SOAP menu and select “Send request to server.” The SOAP response will be displayed.

If authentication is needed (user/pass), a popup window will appear for entering your credentials. It will appear only once until you close the XMLSpy application.

**Pros:** Not only for testing SOAP WS but also for building and editing XML and WSDL; auto-generates the SOAP request from the WSDL file.

**Cons:** Not free

## SAP SOAP Client

The SAP SOAP client tool is free and can be [downloaded from SDN](#).

In order to download it, you need [Java Web Start](#). You will need to configure your Java Web Start to be able to download it if you are using an Internet proxy. Run Java Web Start, go to the file, select “Preferences,” and set your proxy settings. Then try again to download the tool from the link.

In the SAP SOAP client tool, you have to provide the target URL of your WS. It can be found in the WSDL in the `soap:address` XML element. You will also have to provide or write the SOAP request. It is not as convenient as XMLSpy, which generates the SOAP request from the WSDL file for you.

In the menu, go to “View” and select “Setting.” There you can add a proxy setting or the basic authentication user/pass.

Notice that sometimes you can get a response with an HTML code 302. It means that it has been redirected. It can be caused by an authentication need. In some cases, when the user/password authentication is requested, the WS call is redirected to another page.

**Pros:** Free

**Cons:** The SOAP request needs to be written by you.

## Author Bio

Gilles Atlan is a SAP NetWeaver Solution Architect, mainly oriented in J2EE technology.