

# Web Services Testing and SAP NetWeaver Application Server, Java EE 5 Edition

## Applies to:

SAP NetWeaver Application Server, Java EE 5 Edition

## Summary

With the introduction of SAP NetWeaver Application Server Java EE 5 Edition, SAP has moved in the right direction by providing a platform for building enterprise-class, scalable, distributed, and open standards-based Service-Oriented Architecture (SOA). This article describes the process of building a simple Web service using the new SAP NetWeaver Application Server and testing its characteristics using a Web services testing tool.

**Author(s):** Mamoon Yunus

**Company:** Crosscheck Networks, Inc.

**Created on:** October 15, 2006

## Author Bio



Mamoon Yunus is an Advisor for Crosscheck Networks, CTO of Forum Systems and a pioneer of Web services-based SOA Gateways & Firewalls. Prior to Forum, Mr. Yunus was at webMethods where he developed XML-based technology. Mamoon holds two Graduate Degrees in Engineering from MIT.

## Table of Contents

Introduction .....	3
Step 1: Download and Install SAP NetWeaver Application Server, Java EE 5 Edition.....	3
Step 2: Download and Install SOAPSonar .....	4
Step 3: Develop a Simple Web Service Using SAP NetWeaver Developer Studio .....	4
Step 4: Load and Test the Web Service.....	8
Conclusions .....	9
Related Content.....	9
Disclaimer and Liability Notice.....	10

## Introduction

With the introduction of SAP NetWeaver Application Server Java EE 5 Edition, SAP has moved in the right direction by providing a platform for building enterprise-class, scalable, distributed and open standards-based Service Oriented Architecture (SOA). SAP NetWeaver Application Server, Java EE 5 Edition is one of the first application servers in the market to achieve [Java EE 5 compatibility](#). With such leadership in adopting standards-based specifications, the perception of SAP as a closed, monolithic and conservative behemoth unwilling to share its internal riches (APIs) with common developers has been shattered. The preview release of its new Java EE 5-based Application Server and its first-to-market strategy in adopting developer-friendly specifications shows SAP's desire to appeal to a broader base of developers.

The primary focus of Java EE 5 is to transition J2EE to a more developer-friendly architecture. One of more exciting parts of Java EE 5 specification is a heavy support for annotations that enable developers to cleanly expose already-developed [Java code as Web services through meta-data](#) for classes or methods.

This article describes the process of building a simple HelloWorld Web service using SAP NetWeaver and testing its characteristics using a simple Web services testing tool - SOAPSonar Enterprise Edition. SOAPSonar™ is a Web services testing product that provides comprehensive testing across complex, dependent and distributed deployments.

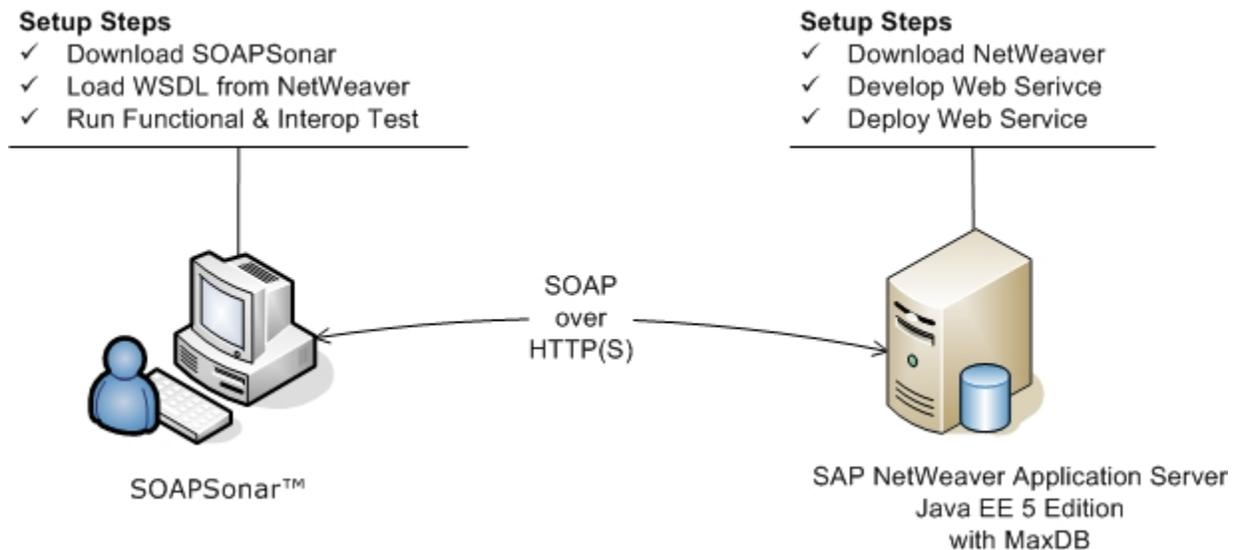


Figure 1: Web Services Testing NetWeaver Java EE 5 Edition using SOAPSonar.

As you build a robust, secure, and interoperable Web services-enabled Service-Oriented Architecture (SOA) with NetWeaver as a core component, using products like SOAPSonar to establish the quality and repeatability of your Web services-based IT assets will become crucial.

### Step 1: [Download and Install SAP NetWeaver Application Server, Java EE 5 Edition](#)

Download SAP NetWeaver Application Server, Java EE 5 Edition from the SAP Developer Network (SDN). Registration is required, but is simple and painless. Once you complete the registration process, you may have to refresh your browser if you are using Mozilla Firefox. The download is about 650 MB and includes trial versions of the SAP NetWeaver Application Server, MaxDB 7.6 and the SAP NetWeaver Developer Studio. System requirements for installation are as follows:

- 512 MB RAM (1 GB recommended).
- Intel Pentium III/1.1GHz or higher.
- 2GB hard disk space.

The file is in RAR format and can be unpacked using your favorite RAR tool. Once all the files are unpacked, simply click on the setup executable to start the installation process. Before starting the installation process, it is highly recommended to shut down all applications, firewalls, VNC services, and any service that may be holding on to a port required by SAP NetWeaver.

If you come across any installation issues, use the [SDN Web AS Preview Installation Forum](#). The participants are very helpful and responsive. For a good overview of pre-installation steps, see Oliver Schmidt's blog entry, [Installing and operating the SAP NetWeaver J2EE Preview](#).

## Step 2: [Download and Install SOAPSonar](#)

SOAPSonar Enterprise Edition can be downloaded from [here](#). Registration is required because this product is subject to United States export laws. SOAPSonar can be installed on Windows XP/2K3 platforms.

SOAPSonar can easily consume WSDLs generated by SAP NetWeaver and enable SOAP messaging as shown in Figure 1 above. You can load as many WSDLs in the Web services testing tool as you wish and mix and match tests across different WSDL operations. All test projects can be saved to your desktop. SOAPSonar also provides operation chaining, so you can take the output of an operation and use it as the input for another operation. Once you develop and deploy the Web service as described in Step 3, you will be able to begin SOAP messaging between SOAPSonar and SAP NetWeaver.

## Step 3: Develop a Simple Web Service Using SAP NetWeaver Developer Studio

SAP NetWeaver Developer Studio is packaged as a component of the SAP NetWeaver Application Server Java EE 5 Edition install, so you will not need to install any more applications.

Before starting SAP NetWeaver Developer Studio, you can verify that the SAP NetWeaver Application Server is up and running by navigating a web browser to **http://localhost:50000**. A welcome page will be served if your Application Server is up and running. If the Application Server is not up, the simplest way to start it up is to execute *Start > Programs > SAP NetWeaver > Application Server JP1 > Start Application Server*. You can also use the SAP Management Console to start and shut down the Application Server as needed.

You are now ready to start building a simple Web service using SAP NetWeaver Developer Studio. Start with the following steps:

1. Start SAP NetWeaver Developer Studio by going to *Start > Programs > SAP NetWeaver > SAP NetWeaver Developer Studio > SAP NetWeaver Developer Studio*. Minimize the Welcome Panel.
2. From the top Menu, select *Windows > Show View > Other*. Under the *Server* folder, select *Servers*. This will bring up a Servers panel in the Developer Studio displaying the server and its status.
3. Select *File > New > Project > EJB > EJB Project 3.0*. This will bring up a New EJB 3.0 Project panel. Enter `HelloWorldEJB` for the Project Name. Leave the Workspace Location selection as the default value. Finally, select the Add Project to EAR checkbox and enter `HelloWorldEAR` as your *EAR Project Name*. After you click Finish, you should see `HelloWorldEJB` and `HelloWorldEAR` in the Navigation Panel on the right.

- Right click on HelloWorldEJB in the Navigation panel and select *New > Other > EJB > EJB Session Bean 3.0*. Enter the values on the New EJB Session Bean 3.0 Panel as shown in Figure 2. Note that the HelloWorldBean is Stateless, has no interfaces, and is part of the `com.sap.tutorial.javaee` package.

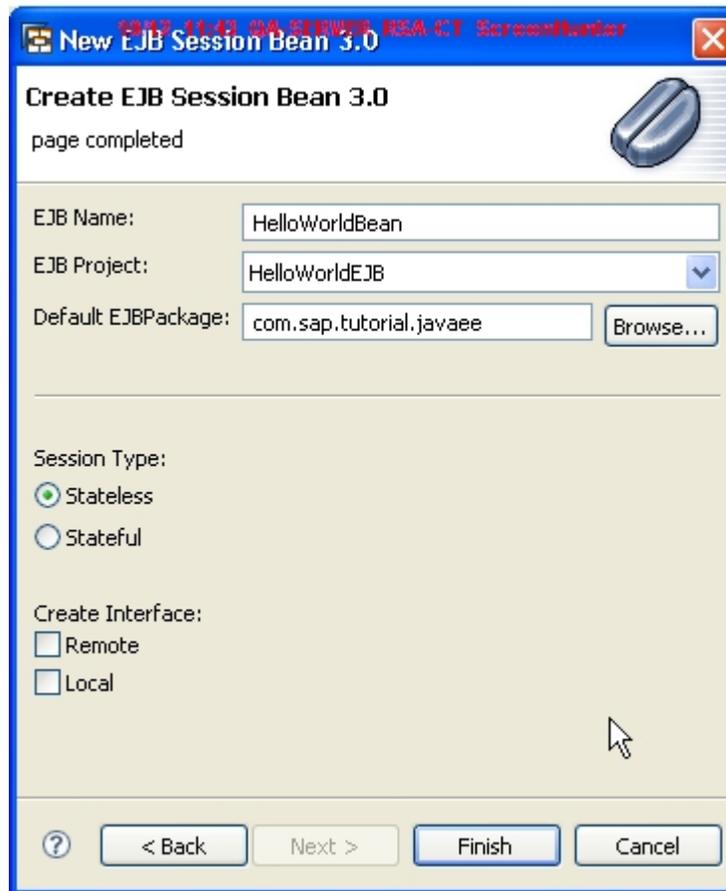


Figure 2: Creating a New Stateless EJB Session Bean - HelloWorldBean

- After you hit Finish in Figure 2, you will see the skeleton code in `HelloWorldBean.java`. You can now type a simple `sayHello(String firstName)` method that returns a `String` as shown in Figure 3 below.

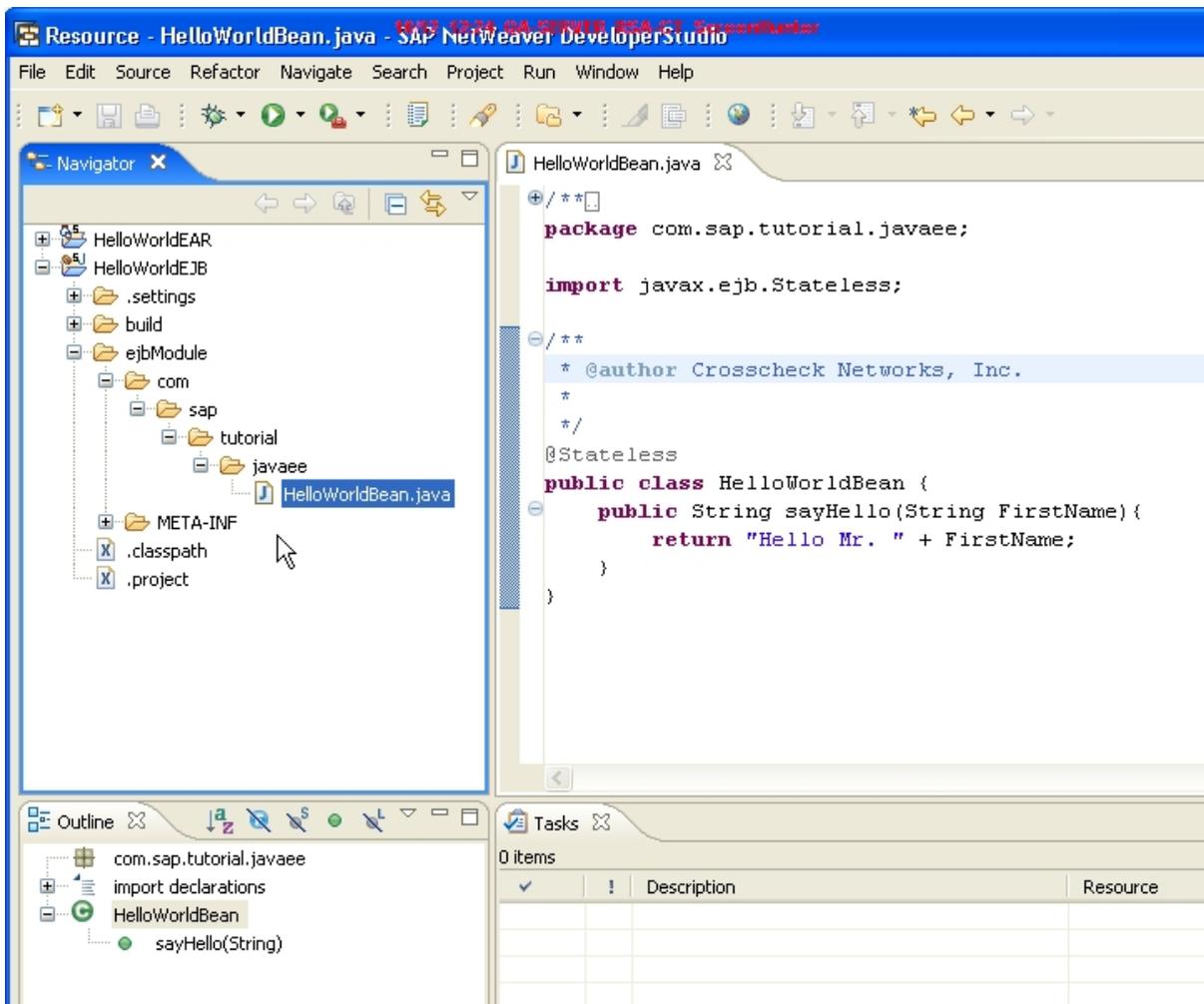


Figure 3: HelloWorldBean Java file with a simple sayHello method.

- Now that we have a simple `HelloWorldBean` ready, SAP NetWeaver can easily convert it into a Web service. Right click on the `HelloWorldBean.java` file highlighted in Figure 3. Select **Web Services > Create Web Service**. The Web Service panel displayed in Figure 3 will appear. Move the left hand slider to change the state to *Develop Service*. Hit Next, and ensure that the `sayHello` method is selected for availability at the endpoint. Once you hit Finish, you will see `@WebService(...)` and `@WebMethods(...)` annotations injected in the `HelloWorldBean.java` file. Voila! Your Web services are developed and ready for deployment.

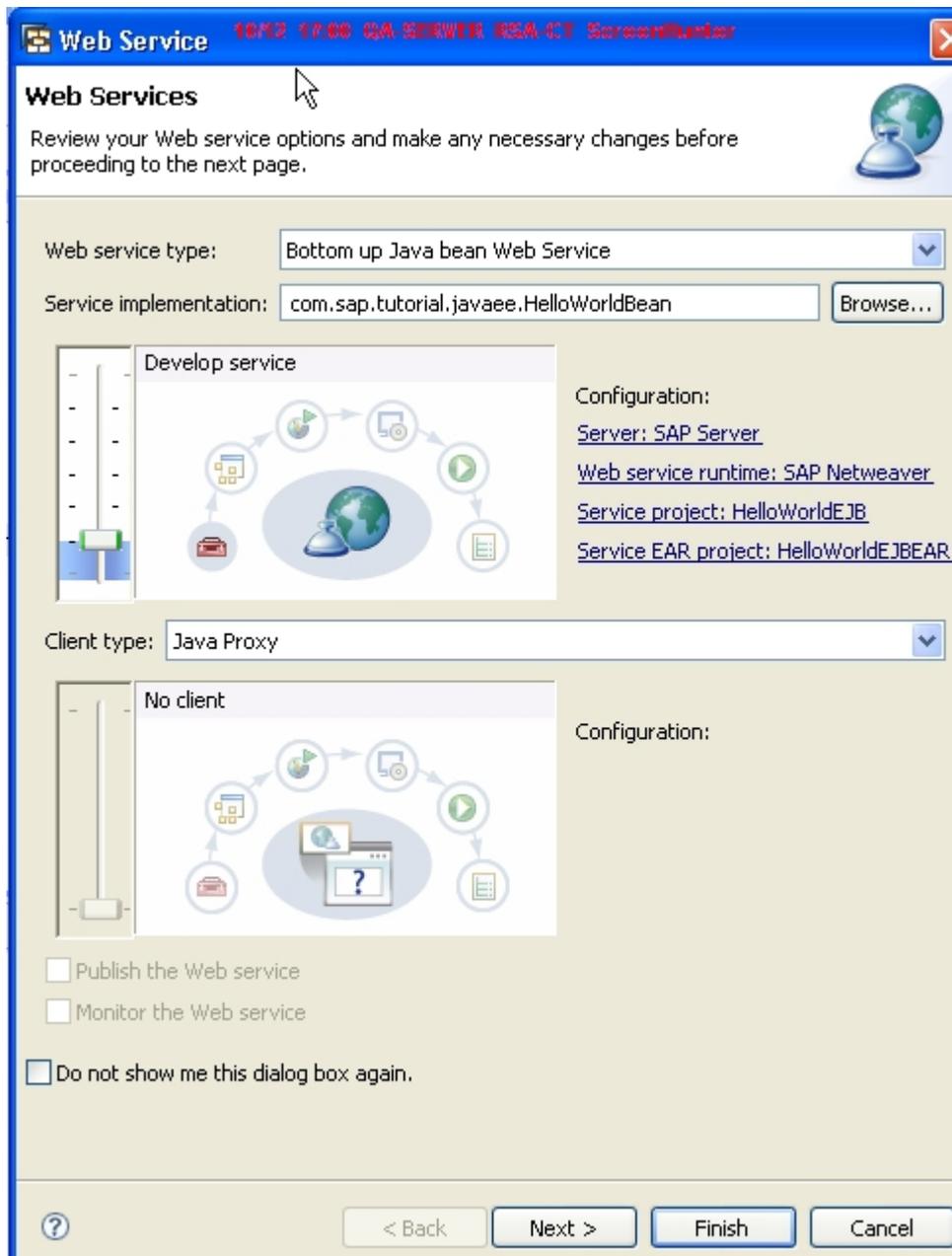


Figure 4: NetWeaver Developer Studio Panel for Web Services enabling HelloWorldBean.

7. The last task is to deploy the Web service. Go to the Servers panel that shows available SAP NetWeaver Application Servers, which as you may recall from task 2 is accomplished by selecting *Windows > Show View > Other* and selecting *Servers*. Right click on the selected server, select *Add and Remove Projects* and move *HelloWorldEAR* from *Available projects* to *Configured projects*.

At this point, the HelloWorldBean is deployed and available for any Web Services client by consuming the service WSDL available at the following endpoint:

http://hostname:50000/HelloWorldBeanService/HelloWorldBean?wsdl

Now we must test this Web service comprehensively by loading it into SOAPSonar.

#### Step 4: Load and Test the Web Service

As shown in Figure 5, the WSDL loaded is loaded into SOAPSonar and the single operation is automatically parsed and displayed in the navigation panel. You can then select the operation and populate the input field values for `arg0`. Hit the arrow key in the Request panel to submit the SOAP request. The response from SAP NetWeaver Application Server is presented in the lower panel and appends *Hello Mr.* to the input value. Please note that NetWeaver adds `<wsse:Security>` tags with a timestamp in the SOAP Header.

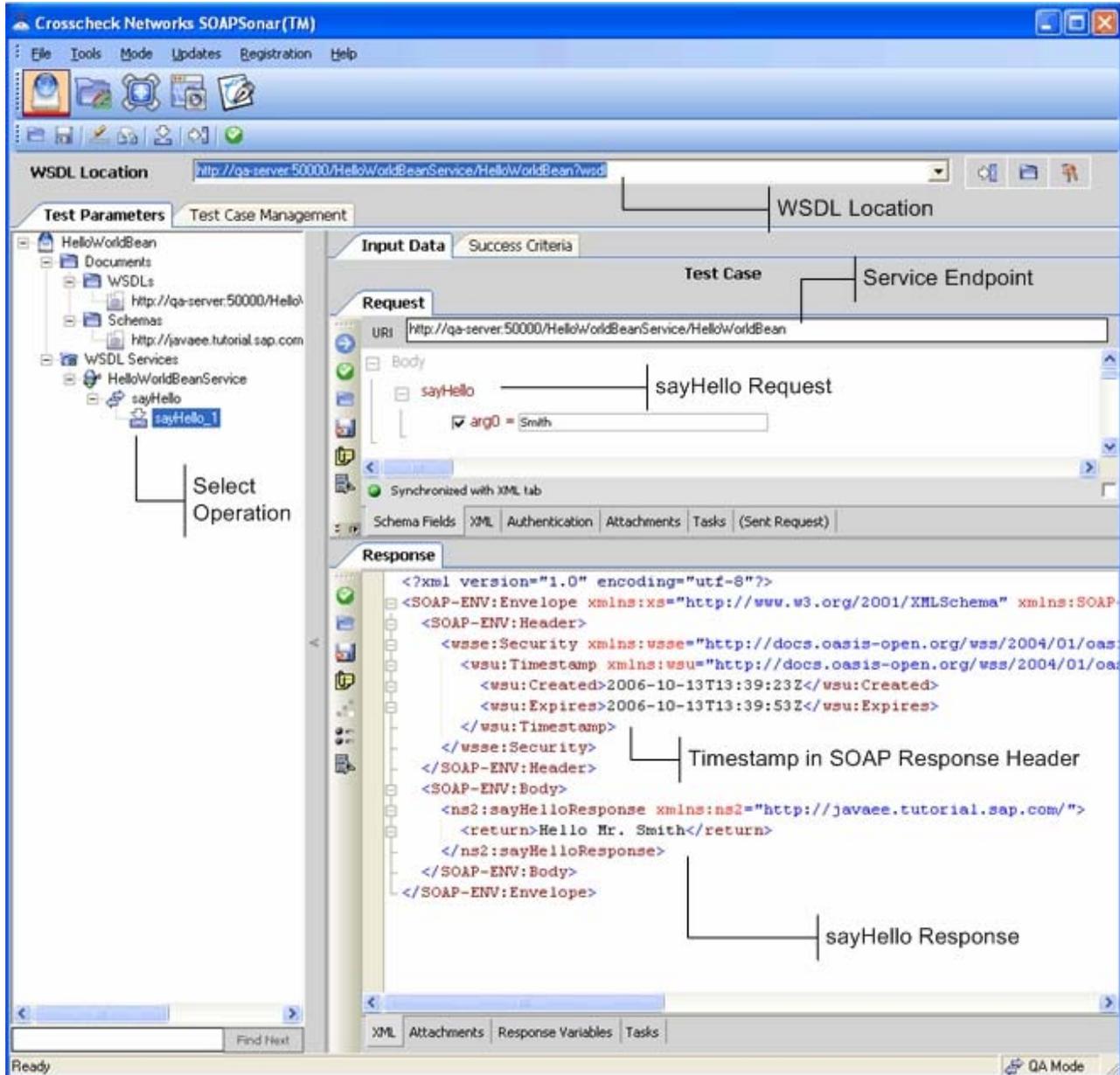


Figure 5: SOAPSonar Web Services Testing Platform for testing NetWeaver Web services.

With SOAPSonar, you can test a published Web service across four areas of testing: Functional Regression, Performance, Interoperability, and Vulnerability Assessment.

**Functional Regression Testing** ensures that the Web service operations function properly over time and that if the underlying code has changed in SAP NetWeaver, then such changes are identified and addressed. SOAPSonar provides the ability to set base-line tests for `sayHello()` and schedule regression tests to ensure that all desired pass/fail test conditions are met. In the sample Web service, the operation `sayHello()` should behave as expected and return a string value.

**Performance Testing** ensures that the Web service operations perform as expected once latency and scalability characteristics are identified.

**Interoperability Evaluation** ensures that the published Web service operations interoperate across various platforms. SOAPSonar provides an additional WS-I Basic Profile 1.1 design-time and run-time test to check for interoperability issues. The design-time interoperability tests should pass, giving you the message "No Failed Assertions Detected." This means that any WS-I Basic Profile-compliant client regardless of application platform such as .NET, Java or LAMP, can readily consume the WSDL generated by SAP NetWeaver.

SOAPSonar also provides the ability to execute stringent run-time tests by sending tests of corner cases. The mutated tests are generated to check whether the endpoint is indeed honoring WS-I Basic Profiles, even though the WSDL itself is WS-I compliant.

**Vulnerability Assessment** ensures that the Web service operations are tested for edge cases and are not vulnerable to information leaks, Denial of Service, or other Web services attacks. With such assessments, vulnerabilities introduced by poor application coding practices, such as lack of exception handling, are quickly determined and action can be taken to remedy the problem. The simple `HelloWorldBean` described in this article is poorly coded with and lacks `try-catch` type constructs to handle exceptions. This should result in an overall Risk Score of 26 with a few Medium Severity issues identified because of stack traces-based information leaks in the response messages.

## Conclusions

Developing Web services in SAP NetWeaver Developer Studio is straightforward and intuitive. The biggest challenge to face is overcoming installation hurdles, and when there is a problem, the SDN forums are readily available. Otherwise, the WSDL generated by NetWeaver is interoperable with .NET-based SOAPSonar and can send SOAP messages between SOAPSonar and SAP NetWeaver Application Server seamlessly.

In our experience, the overall Functional, Performance, Interoperability and Vulnerability profiles for the `HelloWorld` Web service were positive, though there needs to be some tightening of the SOAP handling stack to actively enforce WS-I Basic Profile 1.1 compliance. Also, the corner case or boundary-breaking "Mutant" SOAP messages should be handled elegantly to prevent information probing or inadvertent information leaks left behind by sloppy developers.

As you can see overall, though, SAP NetWeaver Application Server, Java EE 5 Edition is a step in the right direction and provides competitive Web service functionality for sophisticated SOA deployments. Give it a try.

## Related Content

- [Update: An Introduction to Java EE 5 Platform.](#)
- [SAP Developer Network \(SDN\).](#)
- [SDN Web AS Preview Installation Forum](#)
- [Java Edition 5 at SAP](#)

## **Disclaimer and 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.