

Web Service Control Template

Table of contents

1 Introduction.....	2
2 Generating a Service Control.....	2
3 Junit Testing.....	2

1. Introduction

Use the web service control template (located at `<BeehiveRoot>/samples/controls-webservices-blank`) as a starting point for your web service control.

A web service control makes it easy for other applications to become clients of the target web service.

2. Generating a Service Control

To generate a service control you will need a WSDL file from the target web service.

Drop the WSDL file (along with any other schemas required by the web service) into the schemas directory (`<BeehiveRoot>/samples/controls-webservices-blank/schemas`).

Run the ant target `ant build` to generate a JAR for the service control. The build generates the types used by the web service along with the control interfaces and compiles them into a JAR file. The final service control JAR is saved in the build directory.

Note:

Before running the `ant build` target, don't forget to edit the `build.properties` file.

The `controls-webservice-blank` sample contains the WSDL file for the [EnhancedAddressBookWS](#) (`../../wsm/sample_AddressBookEnhanced.html`) sample, but the web service doesn't need to be running to generate its service control.

3. Junit Testing

The `controls-webservice-blank` shows how to write junit tests for the service control, but it is not mandatory to have junit tests for a control.

The junit tests call the web service using the control interfaces generated based on the WSDL. The tests add an entry and retrieve it to verify the web service. To run the junit tests make sure you have the [EnhancedAddressBookWS](#) (`../../wsm/sample_AddressBookEnhanced.html`) (or whichever target web service) up and running.

To run the junit tests run the target: `ant run.junit`

Note:

Controls are meant to be used in a container; in a container (servlet, ejb, spring) the initialization happens transparent to the

Web Service Control Template

application. The junit tests here are standalone and are running outside a container, so we have to do some of the work that the container normally would do. The junit setUp includes the control initialization. In the junit environment this is done in the setUp. Be aware that the in the future we plan to automate the junit initialization for the running controls.

Note:

The build file uses XMLBeans to generate types. XMLBeans should only be used for the Literal use (RPC or Document style) WSDLs. For RPC-Encoded WSDLs Axis tools must be used to generate the types. The build.xml file contains the targets to generate Axis types but they are commented out. To use Axis tools uncomment the Axis related lines and make the necessary changes to the build file.