Axis C++ Client User's Guide

<!-- -->

1. Contents

2. Introduction

In order to use web services you need to create client-side stubs that help you to access the service You then use these stubs within your application.
If you are also responsible for writing the service you need to create the service skeletons and then complete them. This document explains how to use the Axis CPP tooling (WSDL2Ws) to generate and use the client code. If you want to learn how to create your web services please look here.

3. Pre-requisites

WSDL2Ws is a 100% java tool and requires a version of Java to be on the machine that you create your stubs and skeletons on. The version of Java that is required is >= 1.4
WSDL2Ws also has a number of pre-requisite jar files that need to be added to your classpath when you run the tooling
<AxisCPP Install dir>/lib/axis/wsdl2ws.jar: Contains the main WSDL2Ws code.
<AxisCPP Install dir>/lib/axisjava/axis.jar: Contains the Axis java code which WSDL2Ws is based on
<AxisCPP Install dir>/lib/axisjava/commons-discovery.jar:
<AxisCPP Install dir>/lib/axisjava/commons-logging.jar;
<AxisCPP Install dir>/lib/axisjava/jaxrpc.jar;
<AxisCPP Install dir>/lib/axisjava/saaj.jar;
<AxisCPP Install dir>/lib/axisjava/wsdl4j.jar
If you want to learn more about WSDL2Ws please see this reference document.

4. Generating and using client-side stubs

AxisCPP has Java based tooling. If you supply the WSDL which describes your service then

WSDL2Ws will produce client-side stubs for you. All you need to do is create your application that uses those stubs. Throughout this section we will use the Calculator service that comes with Axis as an example of how to use the tooling.

Note: In order to run the application discussed below you need the "calculator" service deployed to a server. If you don't want to deploy the calculator service then you need to follow the instructions below but use your own WSDL files that describe your service. If you do want to deploy the calculator service for Apache web server or the simple_axis_server please go here. Once you've deployed your service then come back to these instructions and run the calculator client..

4.1. Generating Calculator client C++ classes

Firstly copy the Calculator wsdl to the client samples directory e.g. (linux)

cd <Axis installation directory>/samples/client/calculator

cp -f <Axis installation directory>/wsdls/Calculator.wsdl ./

IMPORTANT:In this example we are showing you how to use the WSDL2Ws tooling to generate the stubs using Calculator.wsdl. However, in the <Axis installation directory>/samples/client/calculator folder you will already find generated files. If you wish to use those without generating new ones you can do so. We recommend that you run the calculator sample with the already generated files firstly and later practice using the tooling with Calculator.wsdl.
Next you create the client-side stubs that represent the Calculator service.

Note: Don't forget to add all of the pre-requisite jar files into your classpath

java org.apache.axis.wsdl.wsdl2ws.WSDL2Ws Calculator.wsdl -lc++ -sclient

Note: If you specify **-o<target directory>** you will have source generated inside the specific folder instead of the current folder where the tool is run.
You now have the client-side stubs generated for you. You now need to create a client that uses those stubs. This class contains a main method in which Calculator instance is created and its methods are called.

```
#include "Calculator.h"
#include<stdio.h>
int main()
 {
  Calculator c;
  int intOut;
```

```
 c.add(20, 40, intOut);
 printf("result is = %d\n", intOut);
 return 0;
}
```

Now build your client application with your compiler of choice (in this example we have used g++ on a linux system)

cd <Axis installation directory>/samples/client/calculator

g++ *.cpp -I<Axis installation directory>/include -L<Axis installation directory>/lib -ldl -laxiscpp_client -ocalculator

Note: In order to run your application you need the calculator service deployed to a server. In order to do that for Apache web server or the simple_axis_server please go here. Once you've deployed your service then come back to these instructions and run the calculator service.
In this instance we have deployed the service to localhost. and are asking the calculator service to add 10 and 5 for us.
Running the calculator sample (linux)

./calculator add 10 5 http://localhost/axis/Calculator

Now that you've done the sample here you can go on and create your own services and clients using your own service definitions files (WSDL's).
Thankyou for using Axis CPP :-)