

CS 683 Emerging Technologies
Spring Semester, 2003
Doc 17 Axis & Soap
Contents

AXIS & WSDL 2
WSDL for Axis Soap Service 2
Java From WSDL..... 5
Server Side Code Generation..... 16
How to generate WSDL without an Existing Service..... 21

Reference

Axis User's Guide, <http://ws.apache.org/axis/>

2003 SDSU & Roger Whitney, 5500 Campanile Drive, San Diego, CA 92182-7700 USA.
OpenContent (<http://www.opencontent.org/opl.shtml>) license defines the copyright on this document.

AXIS & WSDL

WSDL for Axis Soap Service

Append ?WSDL to the end of the Soap Service URL

Example Calculator.jws

```
public class Calculator {  
    public int add(int i1, int i2)  
    {  
        return i1 + i2;  
    }  
}
```

```
    public int subtract(int i1, int i2)  
    {  
        return i1 - i2;  
    }  
}
```

Place the file in Tomcat/webapps/axis folder

From local machine service url is:

<http://localhost:8080/axis/Calculator.jws>

So <http://localhost:8080/axis/Calculator.jws?WSDL>

Will return the WSDL

Doing it with Telnet

telnet localhost 8080

Trying ::1...

Connected to localhost.

Escape character is '^]'.
GET /axis/Calculator.jws?WSDL

Result

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="http://localhost:8080/axis/Calculator.jws"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:apachesoap="http://xml.apache.org/xml-soap"
  xmlns:impl="http://localhost:8080/axis/Calculator.jws"
  xmlns:intf="http://localhost:8080/axis/Calculator.jws"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"><wsdl:types/>
<wsdl:message name="subtractRequest">
  <wsdl:part name="i1" type="xsd:int"/>
  <wsdl:part name="i2" type="xsd:int"/>
</wsdl:message>
<wsdl:message name="subtractResponse">
  <wsdl:part name="subtractReturn" type="xsd:int"/>
</wsdl:message>
<wsdl:message name="addResponse">
  <wsdl:part name="addReturn" type="xsd:int"/>
</wsdl:message>
<wsdl:message name="addRequest">
  <wsdl:part name="i1" type="xsd:int"/>
  <wsdl:part name="i2" type="xsd:int"/>
</wsdl:message>
<wsdl:portType name="Calculator">
  <wsdl:operation name="add" parameterOrder="i1 i2">
    <wsdl:input message="impl:addRequest" name="addRequest"/>
    <wsdl:output message="impl:addResponse" name="addResponse"/>
  </wsdl:operation>
</wsdl:portType>
</wsdl:definitions>
```

```
</wsdl:operation>
<wsdl:operation name="subtract" parameterOrder="i1 i2">
  <wsdl:input message="impl:subtractRequest" name="subtractRequest"/>
  <wsdl:output message="impl:subtractResponse" name="subtractResponse"/>
</wsdl:operation>
</wsdl:portType>
<wsdl:binding name="CalculatorSoapBinding" type="impl:Calculator">
  <wsdlsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="add">
    <wsdlsoap:operation soapAction=""/>
    <wsdl:input name="addRequest">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        namespace="http://localhost:8080/axis/Calculator.jws" use="encoded"/>
    </wsdl:input>
    <wsdl:output name="addResponse">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        namespace="http://localhost:8080/axis/Calculator.jws" use="encoded"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="subtract">
    <wsdlsoap:operation soapAction=""/>
    <wsdl:input name="subtractRequest">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        namespace="http://localhost:8080/axis/Calculator.jws" use="encoded"/>
    </wsdl:input>
    <wsdl:output name="subtractResponse">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        namespace="http://localhost:8080/axis/Calculator.jws" use="encoded"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
<wsdl:service name="CalculatorService">
  <wsdl:port binding="impl:CalculatorSoapBinding" name="Calculator">
    <wsdlsoap:address location="http://localhost:8080/axis/Calculator.jws"/>
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

Java From WSDL

`java org.apache.axis.wsdl.WSDL2Java wsdlFile`

generates Java code and stubs for a client from the WSDL

WSDL clause	Java class(es) generated
For each entry in the type section	A java class
	A holder for inout/out parameters
For each portType	A java interface
For each binding	A stub class
For each service	A service interface
	A service implementation (the locator)

Classes generated

- Make it easy to build client
- Typical autogenerated code

From the Calculator WSDL we get 4 classes:

- Calculator.java (331 bytes)
- CalculatorServiceLocator.java (462 bytes)
- CalculatorService.java (3577 bytes)
- CalculatorSoapBindingStub.java (5724 bytes)

Client built using Generated Files

```
package localhost;
```

```
public class Client {
```

```
    public static void main( String[] arguments) throws Exception {  
        CalculatorService addFinder = new CalculatorServiceLocator();
```

```
        //You can specify the location
```

```
        Calculator adder = addFinder.getCalculator(  
            new java.net.URL("http://localhost:8080/axis/Calculator.jws"));
```

```
        System.out.println( adder.add( 1, 2 ));
```

```
        //Or your can just location in the WSDL
```

```
        adder = addFinder.getCalculator();
```

```
        System.out.println( adder.add( 3, 2 ));
```

```
    }
```

```
}
```

Hand Built Client without using Generated Code

```
package samples.userguide.example2 ;

import org.apache.axis.client.Call;
import org.apache.axis.client.Service;
import org.apache.axis.encoding.XMLType;
import org.apache.axis.utils.Options;

import javax.xml.rpc.ParameterMode;

public class CalcClient
{
    public static void main(String [] args) throws Exception {
        Options options = new Options(args);

        String endpoint = "http://localhost:" + options.getPort() +
            "/axis/Calculator.jws";

        args = options.getRemainingArgs();

        if (args == null || args.length != 3) {
            System.err.println("Usage: CalcClient <add|subtract> arg1 arg2");
            return;
        }

        String method = args[0];
        if (!(method.equals("add") || method.equals("subtract"))) {
            System.err.println("Usage: CalcClient <add|subtract> arg1 arg2");
            return;
        }

        Integer i1 = new Integer(args[1]);
        Integer i2 = new Integer(args[2]);
```

```
Service service = new Service();  
Call call = (Call) service.createCall();
```

```
call.setTargetEndpointAddress( new java.net.URL(endpoint) );  
call.setOperationName( method );  
call.addParameter( "op1", XMLType.XSD_INT, ParameterMode.IN );  
call.addParameter( "op2", XMLType.XSD_INT, ParameterMode.IN );  
call.setReturnType( XMLType.XSD_INT );
```

```
Integer ret = (Integer) call.invoke( new Object [] { i1, i2 } );
```

```
System.out.println("Got result : " + ret);
```

```
}  
}
```


Generated Code

Calculator.java

```
package localhost;
```

```
public interface Calculator extends java.rmi.Remote {  
    public int add(int i1, int i2) throws java.rmi.RemoteException;  
    public int subtract(int i1, int i2) throws java.rmi.RemoteException;  
}
```

CalculatorService.java

```
package localhost;
```

```
public interface CalculatorService extends javax.xml.rpc.Service {  
    public java.lang.String getCalculatorAddress();  
  
    public localhost.Calculator getCalculator() throws  
        javax.xml.rpc.ServiceException;  
  
    public localhost.Calculator getCalculator(java.net.URL  
portAddress)  
        throws javax.xml.rpc.ServiceException;  
}
```

CalculatorServiceLocator.java

```
/**
 * CalculatorServiceLocator.java
 *
 * This file was auto-generated from WSDL
 * by the Apache Axis WSDL2Java emitter.
 */

package localhost;

public class CalculatorServiceLocator extends org.apache.axis.client.Service implements
localhost.CalculatorService {

    // Use to get a proxy class for Calculator
    private final java.lang.String Calculator_address = "http://localhost:8080/axis/Calculator.jws";

    public java.lang.String getCalculatorAddress() {
        return Calculator_address;
    }

    // The WSDD service name defaults to the port name.
    private java.lang.String CalculatorWSDDServiceName = "Calculator";

    public java.lang.String getCalculatorWSDDServiceName() {
        return CalculatorWSDDServiceName;
    }

    public void setCalculatorWSDDServiceName(java.lang.String name) {
        CalculatorWSDDServiceName = name;
    }

    public localhost.Calculator getCalculator() throws javax.xml.rpc.ServiceException {
        java.net.URL endpoint;
        try {
            endpoint = new java.net.URL(Calculator_address);
        }
        catch (java.net.MalformedURLException e) {
            return null; // unlikely as URL was validated in WSDL2Java
        }
        return getCalculator(endpoint);
    }
}
```

```

    public localhost.Calculator getCalculator(java.net.URL portAddress) throws
javax.xml.rpc.ServiceException {
    try {
        localhost.CalculatorSoapBindingStub _stub = new
localhost.CalculatorSoapBindingStub(portAddress, this);
        _stub.setPortName(getCalculatorWSDDServiceName());
        return _stub;
    }
    catch (org.apache.axis.AxisFault e) {
        return null;
    }
}

/**
 * For the given interface, get the stub implementation.
 * If this service has no port for the given interface,
 * then ServiceException is thrown.
 */
public java.rmi.Remote getPort(Class serviceEndpointInterface) throws
javax.xml.rpc.ServiceException {
    try {
        if (localhost.Calculator.class.isAssignableFrom(serviceEndpointInterface)) {
            localhost.CalculatorSoapBindingStub _stub = new localhost.CalculatorSoapBindingStub(new
java.net.URL(Calculator_address), this);
            _stub.setPortName(getCalculatorWSDDServiceName());
            return _stub;
        }
    }
    catch (java.lang.Throwable t) {
        throw new javax.xml.rpc.ServiceException(t);
    }
    throw new javax.xml.rpc.ServiceException("There is no stub implementation for the interface: " +
(serviceEndpointInterface == null ? "null" : serviceEndpointInterface.getName()));
}

/**
 * For the given interface, get the stub implementation.
 * If this service has no port for the given interface,
 * then ServiceException is thrown.
 */
public java.rmi.Remote getPort(javax.xml.namespace.QName portName, Class
serviceEndpointInterface) throws javax.xml.rpc.ServiceException {
    java.rmi.Remote _stub = getPort(serviceEndpointInterface);
    ((org.apache.axis.client.Stub) _stub).setPortName(portName);
    return _stub;
}

```

```
public javax.xml.namespace.QName getServiceName() {
    return new javax.xml.namespace.QName("http://localhost:8080/axis/Calculator.jws",
"CalculatorService");
}

private java.util.HashSet ports = null;

public java.util.Iterator getPorts() {
    if (ports == null) {
        ports = new java.util.HashSet();
        ports.add(new javax.xml.namespace.QName("Calculator"));
    }
    return ports.iterator();
}
}
```

CalculatorSoapBindingStub.java

```
/**
 * CalculatorSoapBindingStub.java
 *
 * This file was auto-generated from WSDL
 * by the Apache Axis WSDL2Java emitter.
 */

package localhost;

public class CalculatorSoapBindingStub extends org.apache.axis.client.Stub implements
localhost.Calculator {
    private java.util.Vector cachedSerClasses = new java.util.Vector();
    private java.util.Vector cachedSerQNames = new java.util.Vector();
    private java.util.Vector cachedSerFactories = new java.util.Vector();
    private java.util.Vector cachedDeserFactories = new java.util.Vector();

    public CalculatorSoapBindingStub() throws org.apache.axis.AxisFault {
        this(null);
    }

    public CalculatorSoapBindingStub(java.net.URL endpointURL, javax.xml.rpc.Service service) throws
org.apache.axis.AxisFault {
        this(service);
        super.cachedEndpoint = endpointURL;
    }
}
```

```
public CalculatorSoapBindingStub(javax.xml.rpc.Service service) throws org.apache.axis.AxisFault {
    if (service == null) {
        super.service = new org.apache.axis.client.Service();
    } else {
        super.service = service;
    }
}

private org.apache.axis.client.Call createCall() throws java.rmi.RemoteException {
    try {
        org.apache.axis.client.Call _call =
            (org.apache.axis.client.Call) super.service.createCall();
        if (super.maintainSessionSet) {
            _call.setMaintainSession(super.maintainSession);
        }
        if (super.cachedUsername != null) {
            _call.setUsername(super.cachedUsername);
        }
        if (super.cachedPassword != null) {
            _call.setPassword(super.cachedPassword);
        }
        if (super.cachedEndpoint != null) {
            _call.setTargetEndpointAddress(super.cachedEndpoint);
        }
        if (super.cachedTimeout != null) {
            _call.setTimeout(super.cachedTimeout);
        }
        if (super.cachedPortName != null) {
            _call.setPortName(super.cachedPortName);
        }
        java.util.Enumeration keys = super.cachedProperties.keys();
        while (keys.hasMoreElements()) {
            java.lang.String key = (java.lang.String) keys.nextElement();
            if (_call.isPropertySupported(key))
                _call.setProperty(key, super.cachedProperties.get(key));
            else
                _call.setScopedProperty(key, super.cachedProperties.get(key));
        }
        return _call;
    }
    catch (java.lang.Throwable t) {
        throw new org.apache.axis.AxisFault("Failure trying to get the Call object", t);
    }
}
```

```
public int add(int i1, int i2) throws java.rmi.RemoteException {
    if (super.cachedEndpoint == null) {
        throw new org.apache.axis.NoEndPointException();
    }
    org.apache.axis.client.Call _call = createCall();
    _call.addParameter(new javax.xml.namespace.QName("", "i1"), new
javax.xml.namespace.QName("http://www.w3.org/2001/XMLSchema", "int"), int.class,
javax.xml.rpc.ParameterMode.IN);
    _call.addParameter(new javax.xml.namespace.QName("", "i2"), new
javax.xml.namespace.QName("http://www.w3.org/2001/XMLSchema", "int"), int.class,
javax.xml.rpc.ParameterMode.IN);
    _call.setReturnType(new javax.xml.namespace.QName("http://www.w3.org/2001/XMLSchema",
"int"), int.class);
    _call.setUseSOAPAction(true);
    _call.setSOAPActionURI("");
    _call.setOperationStyle("rpc");
    _call.setOperationName(new
javax.xml.namespace.QName("http://localhost:8080/axis/Calculator.jws", "add"));

    java.lang.Object _resp = _call.invoke(new java.lang.Object[] {new java.lang.Integer(i1), new
java.lang.Integer(i2)});

    if (_resp instanceof java.rmi.RemoteException) {
        throw (java.rmi.RemoteException)_resp;
    }
    else {
        try {
            return ((java.lang.Integer) _resp).intValue();
        } catch (java.lang.Exception _exception) {
            return ((java.lang.Integer) org.apache.axis.utils.JavaUtils.convert(_resp, int.class)).intValue();
        }
    }
}
```

```
public int subtract(int i1, int i2) throws java.rmi.RemoteException {
    if (super.cachedEndpoint == null) {
        throw new org.apache.axis.NoEndPointException();
    }
    org.apache.axis.client.Call _call = createCall();
    _call.addParameter(new javax.xml.namespace.QName("", "i1"), new
javax.xml.namespace.QName("http://www.w3.org/2001/XMLSchema", "int"), int.class,
javax.xml.rpc.ParameterMode.IN);
    _call.addParameter(new javax.xml.namespace.QName("", "i2"), new
javax.xml.namespace.QName("http://www.w3.org/2001/XMLSchema", "int"), int.class,
javax.xml.rpc.ParameterMode.IN);
    _call.setReturnType(new javax.xml.namespace.QName("http://www.w3.org/2001/XMLSchema",
"int"), int.class);
    _call.setUseSOAPAction(true);
    _call.setSOAPActionURI("");
    _call.setOperationStyle("rpc");
    _call.setOperationName(new
javax.xml.namespace.QName("http://localhost:8080/axis/Calculator.jws", "subtract"));

    java.lang.Object _resp = _call.invoke(new java.lang.Object[] {new java.lang.Integer(i1), new
java.lang.Integer(i2)});

    if (_resp instanceof java.rmi.RemoteException) {
        throw (java.rmi.RemoteException)_resp;
    }
    else {
        try {
            return ((java.lang.Integer) _resp).intValue();
        } catch (java.lang.Exception _exception) {
            return ((java.lang.Integer) org.apache.axis.utils.JavaUtils.convert(_resp, int.class)).intValue();
        }
    }
}
}
```

Server Side Code Generation

WSDL2Java can be used to generate server side code also

Use

```
java org.apache.axis.wsdl.WSDL2Java --server-side --skeletonDeploy true wsdlFile
```

Additional files generated

WSDL clause	Java class(es) generated
For each binding	A skeleton class
	An implementation template class
For all services	One deploy.wsdd file
	One undeploy.wsdd file

Calculator Example

Additional Files Generated

CalculatorSoapBindingImpl.java

CalculatorSoapBindingSkeleton.java

deploy.wsdd

undeploy.wsdd

To Build the Server

- Implement the methods in CalculatorSoapBindingImpl.java
- Compile code
- Place code in correct location for Tomcat
- Register service using deploy.wsdd

Example

CalculatorSoapBindingImpl as generated

```
/**
 * CalculatorSoapBindingImpl.java
 *
 * This file was auto-generated from WSDL
 * by the Apache Axis WSDL2Java emitter.
 */

package localhost;

public class CalculatorSoapBindingImpl implements
    localhost.Calculator{
    public int add(int i1, int i2) throws java.rmi.RemoteException {
        return -3;
    }

    public int subtract(int i1, int i2) throws java.rmi.RemoteException {
        return -3;
    }
}
```

After Editing

```
package localhost;
```

```
public class CalculatorSoapBindingImpl implements
```

```
    localhost.Calculator{
```

```
    public int add(int i1, int i2) throws java.rmi.RemoteException {
```

```
        return i1 + i2;
```

```
    }
```

```
    public int subtract(int i1, int i2) throws java.rmi.RemoteException {
```

```
        return i1 - i2;
```

```
    }
```

```
}
```

Compile

Compile

- Calculator.java
- CalculatorSoapBindingSkeleton.java
- CalculatorSoapBindingImpl.java

Move the classes to TOMCAT/common/classes/localhost

Note: the classes were put in the localhost package from the WSDL

This should have been changed to a more reasonable package name

Register the service

Run

```
java org.apache.axis.client.AdminClient deploy.wsdd
```

in the directory containing the deploy.wsdd generated for the service

Now use client to access the service

Generating the WSDL

The above example

Started with an existing Soap Service

Generated the WSDL from the service

From the WSDL generated a client and server

How to generate WSDL without an Existing Service

- Generate a Java interface or class for the service
- Create WSDL using Java2WSDL
- Create Client/Server classes using WSDL2Java

Example

AddInterface.java

```
public interface AddInterface {  
    public int add(int a, int b);  
}
```

Compile the interface

```
javac AddInterface.java
```

Generate WSDL

```
java org.apache.axis.wsdl.Java2WSDL  
-l"http://localhost:8080/axis/add"  
-n "urn:cs683" -p"add" "urn:cs683" AddInterface
```

-l indicates address of service

-n indicates the namespace

-p give the mapping between urn and package

See <http://cvs.apache.org/viewcvs.cgi/~checkout~/xml-axis/java/docs/reference.html> for more options

Generate Client/Server base code

```
java org.apache.axis.wsdl.WSDL2Java  
  --server-side --skeletonDeploy true add.wsdl
```

This generates in directory cs683

```
AddInterface.java  
AddSoapBindingImpl.java  
AddInterfaceService.java  
AddSoapBindingSkeleton.java  
AddInterfaceServiceLocator.java  
AddSoapBindingStub.java  
deploy.wsdd  
undeploy.wsdd
```

Implement Server

Edit AddSoapBindingImpl.java so it is:

```
package cs683;
```

```
public class AddSoapBindingImpl implements cs683.AddInterface{  
    public int add(int in0, int in1) throws java.rmi.RemoteException {  
        return in0 + in;  
    }  
}
```


Implement Client

Create cs683/Client.java and edit to be:

```
package cs683;

public class Client {
    public static void main( String[] arguments) throws Exception {
        AddInterfaceService addFinder =
            new AddInterfaceServiceLocator();
        AddInterface adder = addFinder.getadd( );
        System.out.println( adder.add( 1, 2 ));
    }
}
```

Compile Code

In cs683 do:

```
javac *.java
```

Install Server Code

Copy

- AddInterface.class
- AddSoapBindingSkeleton.class
- AddSoapBindingImpl.class

To TOMCAT/common/classes/cs683

Where TOMCAT is where you installed tomcat

Register the Server

```
java org.apache.axis.client.AdminClient deploy.wsdd
```