

PARSE WSDL TO CREATE MODEL OF WEB SERVICE

The easiest way to make a dynamic web service call is to write a GUI interface to your service that loads in and parses the selected WSDL document. It should then display each method with a set of parameters, where the base classes allow complete manipulation of this, to add specific values and invoke the web service. Something like:

```
WsdParser wsdlParser = new WsdParser();  
WsdModel wsdlModel = wsdlParser.parse(wsdlAddress);
```

The model can then be initialised first by calling:
`wsdlModel.convertMessagesToParameters()`

This call should be made only once. In effect, it will create an empty skeleton structure in the model object, but with all of the required parameter fields present for any method structure that you will use. The different wsdl document parts are stored in related separate model sections. For example, the messages are stored in 'WsdMessage' objects and the types in the 'WsdTypes' object. You can then traverse this object and enter a value for each of the parameters.

There are now methods in the WsdOperation class to select the set of parameters for only 'SOAP', 'Http GET' or 'Http POST', but SOAP is used by the system only. These relate to the constant values of WsdConst.SOAP / GET or POST respectively. A method like:

```
paramList = wsdlOperation.getInParamNames(WsdConst.SOAP)
```

will retrieve the input parameter set for the SOAP method call only. Some further programming is required to complete the task, but it should be much easier now. You can print out the structure to check for bugs. Some wsdl files might not match the port type to the port exactly. If that is the case, there is just one check that will select the first available port that is stored.

INVOKE A WEB SERVICE

The following example shows how a Web Service call might be made:

```
String wsdlAddress = "http://wsdl address/wsdl file.ws?WSDL"; //any address
```

```
//From your GUI
```

```
WsdParser wsdlParser = new WsdParser();
```

```
WsdModel wsdlModel = wsdlParser.parse(wsdlAddress);
```

```
...
```

```
//Ask the User to enter a value for each parameter in the selected operation
```

```
//this might be added to the model as follows
WsdOperation wsdOperation = wsdModel.getOperation("Method Name");
WsdParameter wsdParameter = wsdOperation.getInParamAt("Parameter Name",
parameterIndex);
wsdParameter.addValue("Parameter Value");

...

//For a SOAP Web Service method invocation you might then use
CallObject callObject = new CallObject();
String commProtocol = WsdConst.SOAP / GET or POST;
WebServiceMethodInfo webServiceInfo =
    wsdModel.getWebServiceInfo( wsdOperation.getName());

if (webServiceInfo != null)
{
    //SOAP call
    WebSoapService soapService = new WebSoapService();
    reply = soapService.webServiceCall(commProtocol, wsdOperation,
webServiceInfo);

    //check reply type and status and process as required
}
```