JWSDL Users Guide
Ver. 2.2

Introduction

Welcome to I-Technologies Corp's JWSDL and XML family of open source projects.

What is JWSDL?

JWSDL is a Java framework and API for reading and writing Web Service Definition Language (WSDL) documents. JWSDL uses the XML Java marshaling tools in the ItcWorks open source project (also created by ITC). These XML parsing tools use SAX parsers that provide the fastest possible execution speeds. In addition, adding new WSDL element types is greatly simplified in JWSDL because developers don't have to write code to serialize/de-serialize their added elements. The schema type (in the types section) is also built into JWSDL and uses the ITC JSchema framework to return Schema objects so that code doesn't have to be written to handle that common type element.

The framework consists of:
Java interface classes (defined in javax packages … - one interface for each WSDL type) a set of implementation classes, a WSDL reader (which takes a URL to a WSDL document and produces a Definition object) and a WSDL writer (that takes a Definition object and produces a WSDL document as a String or file on disk). JWSDL is based on the W3cs' WSDL specification 1.1.

Each WSDL xml element has its own Java interface. (i.e., <portType> has a PortType interface). The attributes of a given XML WSDL tag are represented as properties in the interface. JWSDL also parses xml schemas that are found in the ***types*** section of a WSDL document. The Types object can return the Schema object (created by the JSchema Framework also part of ItcWorks).

Audience:

JWSDL is for Java developers that need to incorporate WSDL data into their applications and is an ideal framework to build GUI front end tools with. Its rich API makes it simple to read, modify, or create WSDL documents.

What is in JWSDL?

JWSDL is organized into the following packages:

- javax.wsdl – This package contains the interface classes that define the set of the W3C WSDL 1.1 elements.
- javax.wsdl.extensions – This package contains the Extensibility element wrapper for both user defined WSDL elements as well as

the SOAP, HTTP and MIME extensions defined by the W3C.

- javax.wsdl.extensions.http – This package contains the interfaces for HTTP WSDL extensions
- javax.wsdl.extensions.mime – This package contains the interfaces for MIME WSDL extensions
- javax.wsdl.extensions.soap – This package contains the interfaces for SOAP WSDL extensions
- javax.wsdl.util - This package contains the interfaces for WSDL reader and writer classes

- com.itc.wsdl – This package contains the implementation classes for the interfaces defined in javax.wsdl.
- com.itc.wsdl.extensions – This package contains the implementation classes for the interfaces defined in javax.wsdl.extensions
- com.itc.wsdl.extensions.http – This package contains the implementation classes for the interfaces defined in the javax.wsdl.extensions.http

- com.itc.wsdl.extensions.mime – This package contains the implementation classes for the interfaces defined in the javax.wsdl.extensions.mime
- com.itc.wsdl.extensions.soap – This package contains the implementation classes for the interfaces defined in the javax.wsdl.extensions.soap

- com.itc.wsdl.util – This package contains the implementation classes for the interfaces defined in javax.wsdl.util

JWSDL is packaged in the itcworks.jar. The following jars are also required:

- xercesImpl.jar – xml parsing
- xercesAPIs.jar – xml parsing

Reading WSDL documents:

The following is a code snippet that shows how to read WSDL documents using JWSDL:

```
File fileWsdl new File( "./resources/Stockquote.wsdl");
WSDLReader rdr = WSDLFactory.getInstance().newReader();

// Read and parse into framework
Definition def = rdr.readWsdl( fileWsdl.toURL() );

// Extract schema object from the types object.
Schema schema = def.getTypes().getSchema();

// Get a List of all messages defined for this WSDL document
List listMessage = def.getMessages();
```

Writing WSDL documents:

The following is a code snippet that shows how to write XML schema documents using JWSDL:

```
File fileWSDL = new File(“ ./resources/Stockquote.wsdl” );
writer.writeWsdl( def, fileWSDL );

or

String strWsdl = def.toString();
```

For complete examples of using the JWSDL framework, please see the JUnit tester class: test.itc.wsdl.ItcWSDLTester in the JUnitForItc source archive.

