

# Java Web Services Developer Pack



Copyright 2003 David A. Wilson.  
All rights reserved.

# Objectives

---

- Configure to use JWSDP
- Find the right sample program
  - Many in JWSDP
  - More in the Web Services Tutorial
- Find the appropriate documentation
- Use Ant as a build tool

# Web Services Developer Pack

---

- Java XML Pack:
  - Java API for XML Messaging (**JAXM**)
  - Soap with Attachments API for Java (**SAAJ**)
  - Java API for XML Processing (**JAXP**)
  - Java API for XML Registries (**JAXR**)
  - Java API for XML-based RPC (**JAX-RPC**)
- JSP Standard Tag Library (**JSTL**)
- **Tomcat** (Java servlet and JSP container)
  - Administration Tool
  - Web Application Manager
- **Ant** build tool
- **deploytool** Web application deployment utility
- **Registry Server**

# Java Web Services Tutorial

- Introduction

- Understanding XML
- Getting Started With Tomcat
- Web Applications
- JAXP
- SAX
- DOM
- XSLT

- Java APIs

- JAX-RPC
- JAXM
- JAXR

- Web Technologies

- Servlets
- JSP
  - JavaBeans components
  - Custom Tags
  - Standard Tag Library
- Security

- Case Study

- Coffee Break App

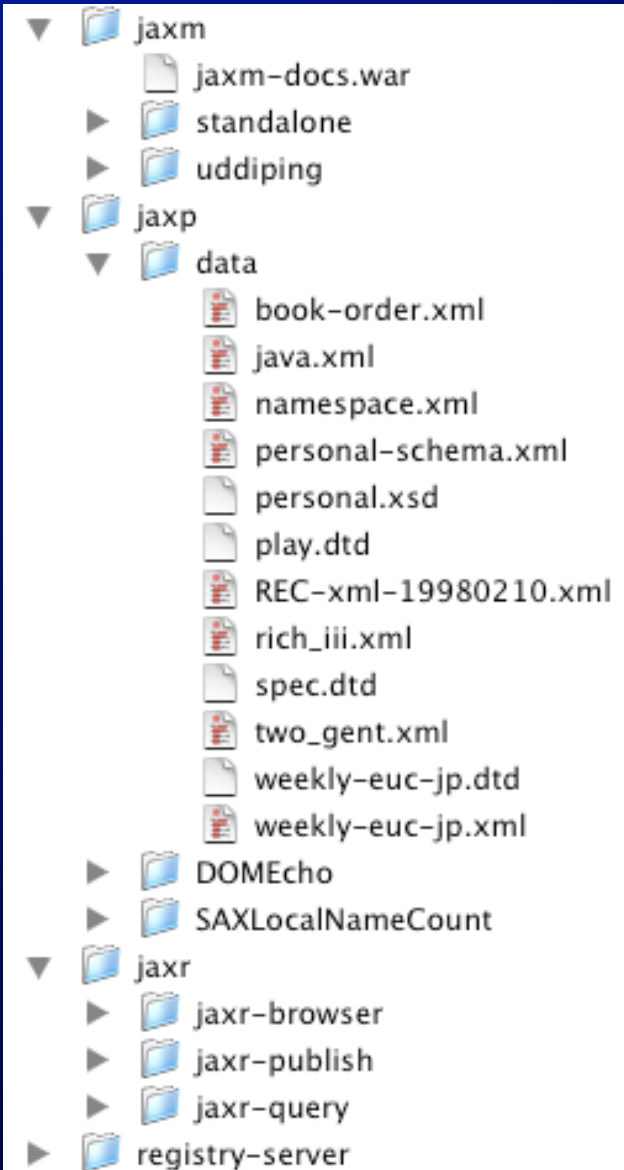
- Appendices

- Tomcat
- JAXM Provider Admin
- xrpc
- Registry Server
- Registry Browser

867 pages

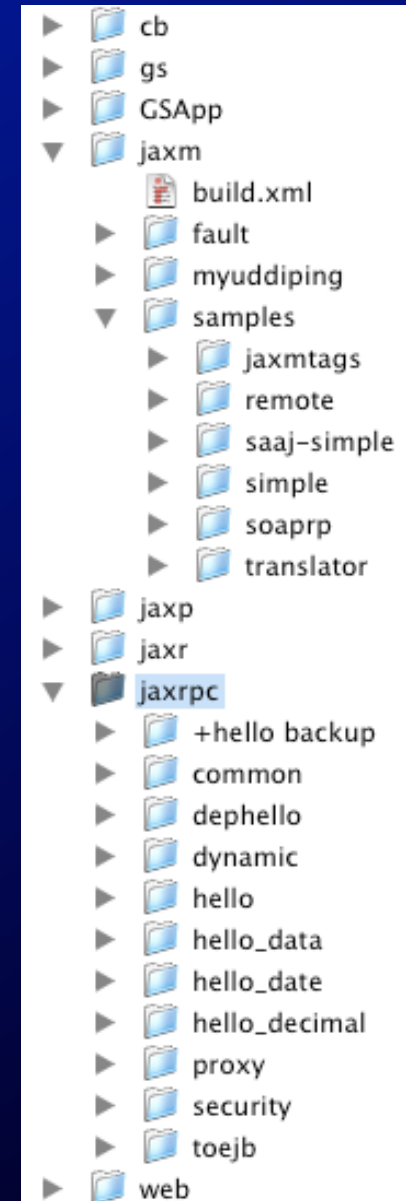
# Samples: JWSDP

- Most samples are built and deployed using Ant.



# Samples: Web Services Tutorial

- cb - Coffee Break
- gs - ConverterBean source
- GSApp - currency converter
- jaxm
- jaxp
- jaxr
- jaxrpc
- web
  - servlets, JSPs

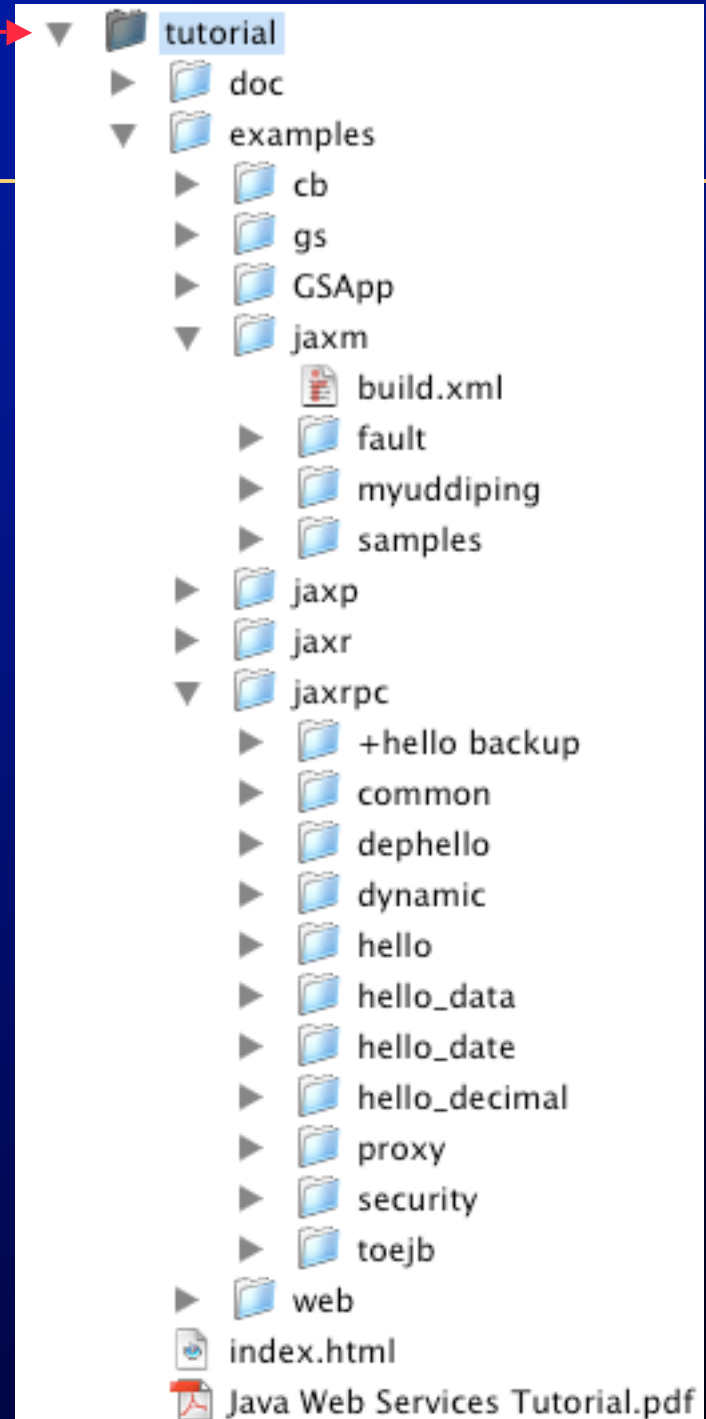
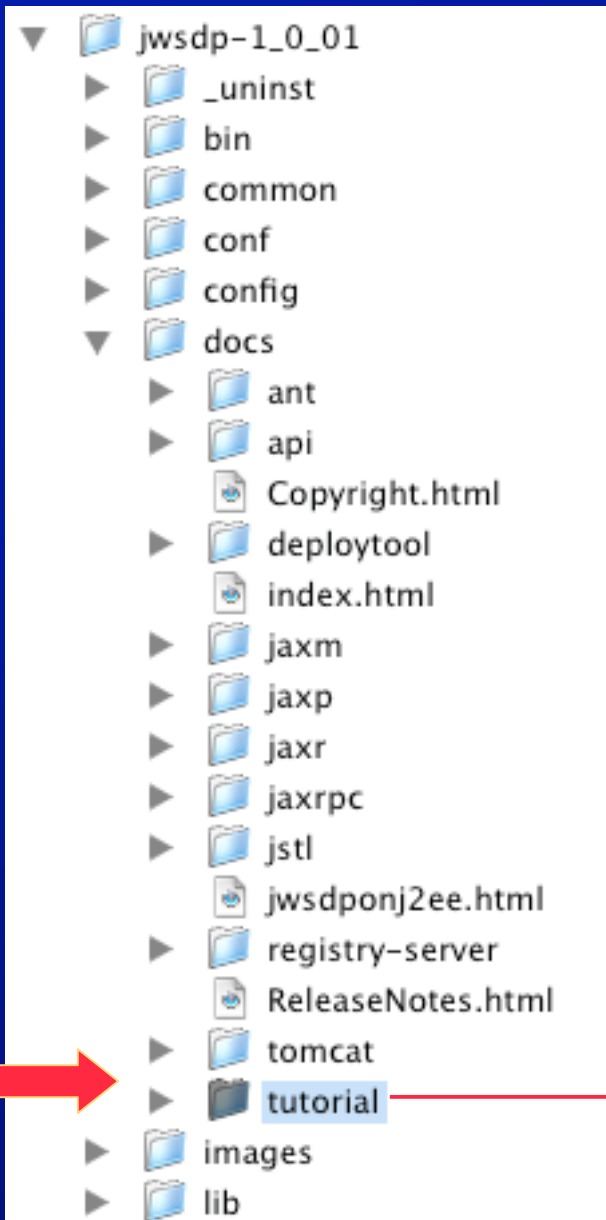


# Java WSDP Configuration

---

- Download and unzip
  - JDK 1.4.x
  - JWSDP
  - Java Web Services Tutorial
- Add to front of **PATH**
  - add bin directory of JWSDP
  - add bin directory of J2SE 1.4
- Create **build.properties** file in home directory
  - `username=your_user_name`
  - `password=your_password`

# Tutorial Files



# Start Tomcat

---

- `cd <jwsdp_home>/bin`

- `./startup.sh`

or

- `startup.bat`

- Wait a minute

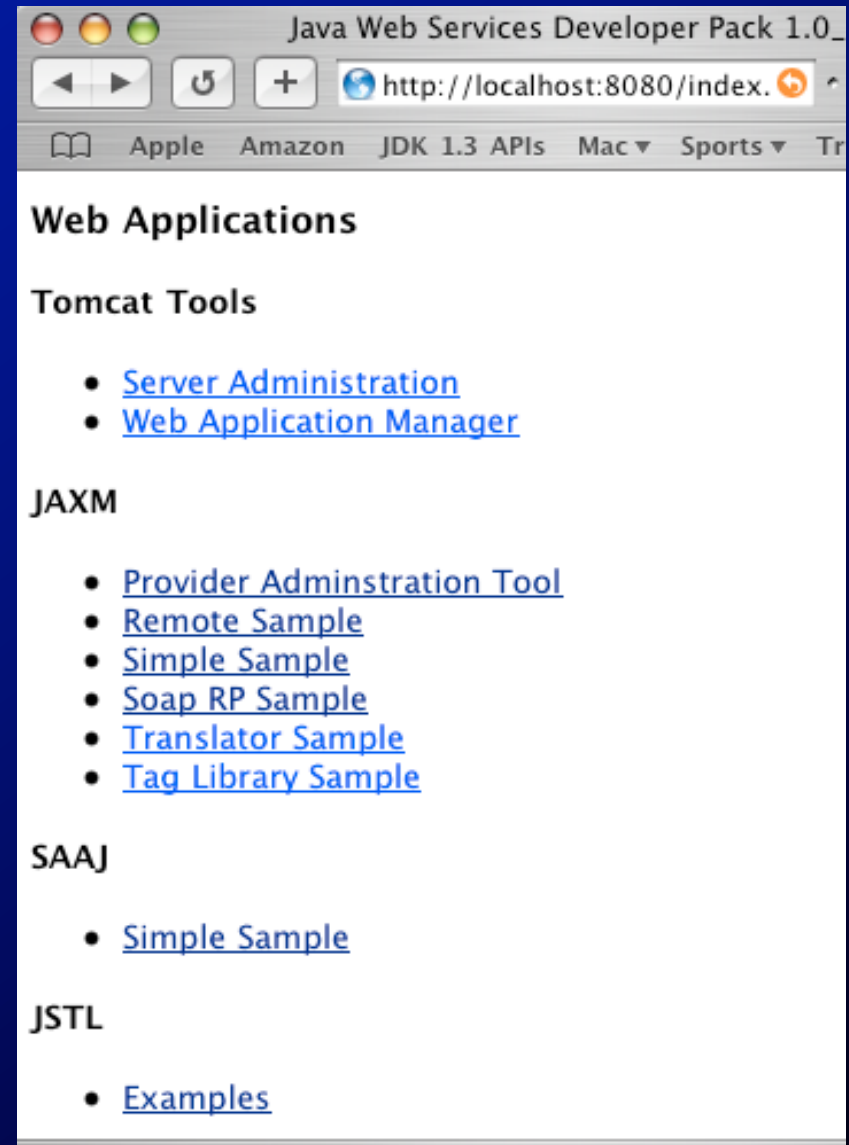
# Compile Initial Samples

---

- `cd <JWSDP_HOME>/docs/tutorial/examples/gs`
- `ant build`
- (make sure to start Tomcat)
- `ant install`

# http://localhost:8080/index.html

- Try these samples from this URL



# Tomcat Server Administration

The image shows two browser windows of the Tomcat Server Administration tool. The left window is the login page at `http://localhost:8080/admin/login.jsp`, featuring a dark red background with the text "TOMCAT WEB SERVER ADMINISTRATION TOOL" and a login form with fields for "User Name" (containing "dwilson") and "Password", and "Login" and "Reset" buttons. The right window is the main administration page at `http://localhost:8080/admin/index.jsp`, which has a light blue header with "TOMCAT WEB SERVER ADMINISTRATION TOOL" and buttons for "Commit Changes" and "Log Out". The main content area includes a tree view of the server configuration, a "Context Properties" table, and a "Loader Properties" table.

**Context Properties**

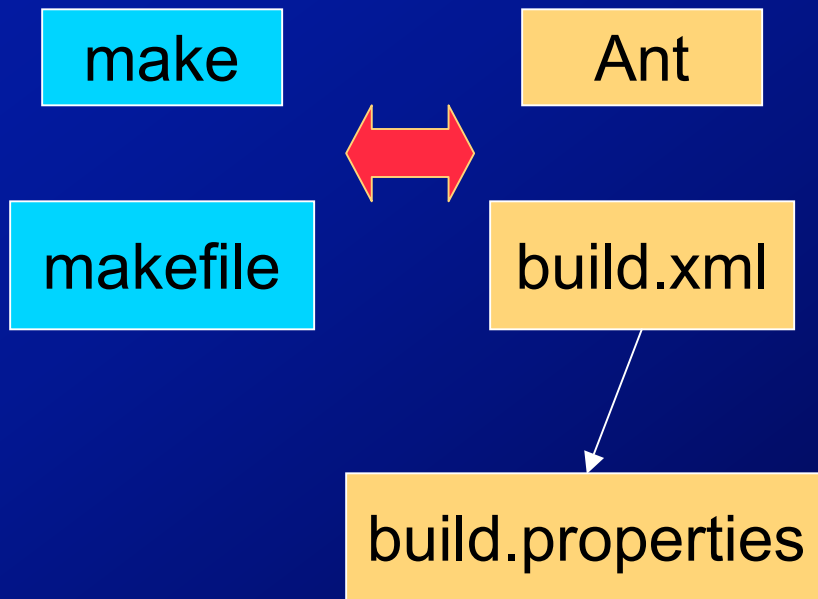
| Property           | Value   |
|--------------------|---|
| Cookies:           | <input checked="" type="checkbox"/> True  |
| Cross Context:     | <input type="checkbox"/> False  |
| Debug Level:       | 0   |
| Document Base:     | /Users/dwilson/dev/jwsdp-1_0_01/work/Standard Engine/localhost/manager/hello-jaxrpc.war |
| Override:          | <input type="checkbox"/> False  |
| Path:              | /hello-jaxrpc   |
| Reloadable:        | <input type="checkbox"/> False  |
| Use Naming:        | <input checked="" type="checkbox"/> True  |
| Working Directory: | work/Standard Engine/localhost/hello-jaxrpc   |

**Loader Properties**

| Property        | Value |
|-----------------|-------|
| Check interval: | 15    |

# Ant

- from Apache Jakarta group
- Build tool for Java
  - analogous to “make”



# Ant

---

- Actions triggered by XML tags
- **Task** - basic unit of work
  - mkdir, javac, ...
- **Target** - group of tasks
  - ant deploy
  - ant run
  - ant deploy run
- **Project** - group of targets
- Many built-in tasks
- Command-line driven
  - GUI under development
  - Integrations exist for popular IDEs

# Ant Built-in Tasks

---

- Ant
- AntCall
- AntStructure
- Apply/  
*ExecOn*
- Available
- Chmod
- Condition
- Copy
- *Copydir*
- *Copyfile*
- Cvs
- CVSPass
- Delete
- *Deltree*
- Dependset
- Ear
- Echo
- Exec
- Fail
- Filter
- Fix
- CRLF
- GenKey
- Get
- GUnzip
- GZip
- Jar
- Java
- Javac
- Javadoc
- Javadoc2
- Mail
- Mkdir
- Move
- Parallel
- Patch
- PathConvert
- Property
- Record
- Rename
- Replace
- Rmic
- Sequential
- SignJar
- Sleep
- Sql
- Style
- Tar
- Taskdef
- Touch
- TStamp
- Typedef
- Unjar
- Untar
- Unwar
- Unzip
- Uptodate
- War
- Zip

# Example build.xml

```
<!DOCTYPE project [  
  <!ENTITY commonTargets SYSTEM "../common/targets.xml">  
>  
  
<project name="JAX-RPC Tutorial" default="build"  
  basedir=".">  
  
  <property file="{user.home}/build.properties"/>  
  <property file="../common/build.properties"/>  
  <property file="build.properties"/>  
  
  &commonTargets; <!-- The ant targets are in  
    ../common/targets.xml -->  
  
  <target name="build" depends="build-service"  
    description="Executes the targets needed to build the  
    service.">  
  </target>  
  
</project>
```

# “targets.xml” (partial)

```
<taskdef name="deploy" classname="org.apache.catalina.ant.DeployTask" />
<taskdef name="list" classname="org.apache.catalina.ant.ListTask" />
<taskdef name="start" classname="org.apache.catalina.ant.StartTask" />
<taskdef name="stop" classname="org.apache.catalina.ant.StopTask" />

<target name="deploy" description="Deploys a Web application">
  <deploy url="${url}" username="${username}" password="${password}"
    path="/${context-path}" war="file:${war-path}"
  />
</target>

<target name="compile-server" depends="prepare"
  description="Compiles the server-side source code">
  <echo message="Compiling the server-side source code...." />
  <javac
    srcdir="."
    destdir="${build}/shared"
    includes="*.java"
    excludes="*Client.java"
  />
</target>
```

# Grouping Targets

---

```
<target name="rebuild"  
  description="Compile and reload web  
              application">  
  <antcall target="build" />  
  <antcall target="remove" />  
  <antcall target="install" />  
</target>
```

# Sample Program Components

---

- **HelloF.java** - service definition interface
- **HelloImpl.java** - service definition implementation
- **HelloClient.java** - remote client
- **config.xml** - configuration file read by *wscompile* tool
- **jaxrpc-ri.xml** - configuration file read by *wsdeploy* tool
- **web.xml** - deployment descriptor for Web component (a servlet) that dispatches to the service
- **build.properties** - read by config.xml

# Cloning a Sample Program

---

- To make a new program, first copy the existing directory, e.g., the “Hello” sample.
- build.properties
  - change example, context-path, client-class
- config.xml
  - change location, packageName
- jaxrpc-ri.xml
  - change displayName, interface, implementation, urlPattern
- Java source files
  - change package name

# Building and Running

---

- **cd** sample\_directory
- **ant clean** (delete old stuff)
- **ant compile-server** (compile server code)
- **ant setup-web-inf** (package war file)
- **ant package** (package war file)
- **ant process-war** (generate WSDL file)
- **ant deploy** (deploy service to Tomcat)
- verify at <http://localhost:8080/goodbye-jaxrpc/goodbye>
- **ant generate-stubs** (used by client)
- **ant compile-client** (compile client)
- **ant jar-client** (create client jar file)
- **ant run** (run client)

# Iterative Development

---

- **cd** sample\_directory
- **ant build** (service)
- **ant redeploy** (service)
- **ant build-static** (client)
- **ant run** (client)

# Resources

---

- WSDP
  - <http://java.sun.com/webservices/webservicespack.html>
- Java Web Services Tutorial
  - <http://java.sun.com/webservices/docs/1.0/tutorial/index.html>
- White Papers
  - <http://java.sun.com/webservices/white/>
- Ant
  - [http://<JWSDP\\_HOME>/docs/ant/manual/index.html](http://<JWSDP_HOME>/docs/ant/manual/index.html)

# White Papers

---

- Web Services Made Easier: The Java APIs & Architectures for XML
  - This paper describes how the various Java™ APIs and Architectures for XML can be used to create Web services.