WebServices - Axis

1. Axis Ant Tasks

Axis comes with Ant tasks to automate aspects of the build process inside ant. To use these tasks, you need

1. Apache Ant 1.5.1 or later
2. The library `axis-ant.jar` which contains the tasks
3. All the main Axis libraries

1.1. Declaring the Tasks

To declare the tasks, set up a classpath to include the axis task JAR and all the dependent libraries. Including everything in the axis lib directory should suffice

```
<path id="axis.classpath">
  <fileset dir="${axis.home}/build/lib">
    <include name="**/*.jar" />
  </fileset>
</path>
```

Then use the `<taskdef>` declaration to declare all the tasks listed in a properties file inside the `axis-ant.jar` file:

```
<taskdef resource="axis-tasks.properties"
  classpathref="axis.classpath" />
```

1.2. Creating Java files from WSDL

This uses the <axis-wsdl2java> task.

```
<axis-wsdl2java
  output="${generated.dir}"
  testcase="true"
  verbose="true"
  url="${local.wsdl}" >
    <mapping
      namespace="http://axis.apache.org/ns/interop"
      package="interop" />
  </axis-wsdl2java>
```

The mapping elements are used to list the mappings from XML namespaces to Java packages; you can include as many as you need.

Copyright © 2000-2005 The Apache Software Foundation All rights reserved.

1.3. Creating WSDL files from Java

This uses the <axis-java2wsdl> task.

1.4. Talking to the admin service

This can be done with the <axis-admin> task.

```
<axis-admin
    port="${target.port}"
    hostname="${target.server}"
    failonerror="true"
    servletpath="${target.appname}/services/AdminService"
    debug="true"
    xmlfile="${endpoint-stub.wsdd}"
    />
```

Here the target.* properties are pulled from a properties file for the system being deployed to, so a single build file can deploy to different systems with ease.

1.5. Foreach Task

<axis-admin>

1.6. Runaxisfunctionaltests Task

<axis-admin>

Copyright © 2000-2005 The Apache Software Foundation All rights reserved.