

Using the Axis2-1.3 Samples

Introduction

A few months ago I started looking at Axis1.4 to get a better understanding of Web Services and after working through a few samples I moved on to Axis2 1.3. Since I had the time, I decided to get a many of the samples working in order to better understand the Axis2 to documentation.

After several weeks of testing and help from the Axis2 community, I got all the samples to work correctly.

This paper shows how I got the Axis2 samples to work, the mistakes I made and the help I received. This document is in no way a replacement for the **13 August 2007 - Apache Axis2/Java Version 1.3 Released** documentation.

Maybe this document may save someone starting out with the Axis2 a lot of unnecessary dead ends.

I read and reread the Axis2 docs and found it concise and complete.

After getting the samples to work, I had a better understanding Axis2.

First and foremost, read the Axis2 documentation and 'Avoiding Mistakes Made Using Axis2 by Deepal Jayasinghe <http://www.developer.com/open/article.php/3589126>.

I do not plan to repeat the Axis2 manual, only the steps that I needed to complete a task that was not described in the manual.

The document will first discuss possible problem areas, followed by the installation of all the necessary tools with simple tests to verify that the installation was OK. The tomcat and axis2 configuration files will be changed in order to make testing easier. Finally, the samples will be reviewed and changes suggested to make the sample work.

Again read the Axis2 getting started, this document assumes that you have read and understand it.

Possible problems to prevent

1. Make sure the Anti-virus checker is not blocking or changing the tcp/ip packets.
For example, Kaspersky 7.0 has an hook into the NDIS driver of the protocol stack and removed the SOAP payload for a connection to Tomcat on the local machine (e.g http\\localhost:8080)
2. Make sure that the firewall is not blocking the following ports:
8080 – tomcat
9080 – when using tcpmon (9080 - 8080)
6060 – maybe if enabled in axis2.xml for SimpleHTTPServer (async)
7060 – tcp client
5001 – soap monitor
3. Backup the `/axis2` context from time to time, because the context might be re-deployed from `axis2.war` or even removed by accident using tomcat undeploy (both happened to me).

The Windows only Installation

Download

1. axis2-1.3-bin.zip
2. axis2-1.3-docs.zip
3. axis2-1.3-war.zip
4. tomcat-5.5.25.zip (the zip, not the windows installer)
5. jdk-1_5_0_13-windows-i586-p.exe (look in Java archive)
6. apache-ant-1.7.0-bin.zip
7. apache-maven-2.0.8-bin.zip
8. tcpmon.jar

Unzip to F:\temp\

```
axis2-1.3-bin.zip      >>> \axis2-1.3\bin \conf,...
axis2-1.3-docs.zip    >>> \axis2-1.3\docs
axis2-1.3-war.zip     >>> axis2.war
tomcat-5.5.25.zip     >>> \apache-tomcat-5.5.25
jdk-1_5_0_13-windows-i586-p.exe >>> installer
apache-ant-1.7.0-bin.zip >>> \apache-ant-1.7.0
apache-maven-2.0.8-bin.zip >>> \apache-maven-2.0.8

tcpmon.jar           >>> not needed
```

Installation and Test

1. Java

- run `jdk-1_5_0_13-windows-i586-p.exe` install in "C:\Program Files\Java\jdk1.5.0_09"
(The blank in the directory will needed to quoted when using a cmd.)

- open a cmd and check the if version and path are correct

```
>java -version
java version "1.5.0_09"
```

- Using a browser open "C:\Program Files\Java\jdk1.5.0_09\demo\applets.html" and select the clock sample to check that the browser handles applets OK.

2. Tomcat

- move `f:\temp\tomcat-5.5.25` to `c:\Temp\Apache-Software-Foundation`¹

```
set CATALINA_BASE=c:\Temp\Apache-Software-Foundation\apache-tomcat-5.5.25
```

```
set CATALINA_HOME=c:\Temp\Apache-Software-Foundation\apache-tomcat-5.5.25
```

```
set PATH=c:\Temp\Apache-Software-Foundation\apache-tomcat-5.5.25\bin;%PATH%
```

(The tomcat `startup.bat` sets these variables, if not defined.)

- open a cmd and start tomcat with

```
>C:\Temp\Apache-Software-Foundation\apache-tomcat-5.5.25\bin\startup.bat
```

- start a browser to <http://localhost:8080/> and see the tomcat icon.

3. Axis2

- move `f:\temp\axis2.war` to
`c:\Temp\Apache-Software-Foundation\apache-tomcat-5.5.25\webapps`

```
set AXIS2_HOME=C:\Temp\Apache-Software-Foundation\apache-tomcat-5.5.25\webapps\axis2\WEB-INF
```

```
set AXIS2_LIB=%AXIS2_HOME%\lib
```

- stop tomcat with `shutdown.bat` and restart tomcat `startup.bat` to install the axis2 context.

The tomcat output should have no errors and look something like

```
[INFO: Deploying web application archive axis2.war
[INFO] Deploying module: addressing-1.3
[INFO] Deploying module: script-1.3
[INFO] Deploying module: metadataExchange-1.3
[INFO] Deploying module: ping-1.3
[INFO] Deploying module: soapmonitor-1.3
[INFO] script module activated

[INFO] Deploying Web service: version-1.3.aar
...
[INFO: Server startup in 10735 ms
```

- start a browser for <http://localhost:8080/axis2> then select "validate" and the following should be displayed:

Found Axis2 default Version service and Axis2 is working properly.

¹ Do not use spaces in the directory or file name, otherwise some of the Axis tools may not find the file.

Hello I am Axis2 version service , My version is 1.3

See the web administration in the axis2 manual.

- stop tomcat and delete the `axis2.war` file to prevent the axis2 context from being overwritten or accidentally un-deployed should the war be deleted while tomcat is running.

4. Ant

- `move f:\temp\apache-ant-1.7.0 to c:\Temp\Apache-Software-Foundation`

`set PATH=C:\Temp\Apache-Software-Foundation\apache-ant-1.7.0\bin;%PATH%`

`set ANT_HOME=C:\Temp\Apache-Software-Foundation\apache-ant-1.7.0`

- open a cmd and start ant

```
>ant -version
Apache Ant version 1.7.0 compiled on December 13 2006
```

5. Maven

- `move f:\temp\apache-maven-2.0.8 to c:\Temp\Apache-Software-Foundation`

`set PATH=C:\Temp\Apache-Software-Foundation\apache-maven-2.0.8\bin;%PATH%`

- open a cmd and start mvn

```
>mvn -version
Maven version: 2.0.8
```

6. Install Axis2 samples

(Overwrite if necessary.)

- `move f:\temp\bin to C:\Temp\Apache-Software-Foundation\jakarta-tomcat-3.3.2\webapps\axis2\WEB-INF`

- `move f:\temp\conf to C:\Temp\Apache-Software-Foundation\jakarta-tomcat-3.3.2\webapps\axis2\WEB-INF`

- `move f:\temp\docs to C:\Temp\Apache-Software-Foundation\jakarta-tomcat-3.3.2\webapps\axis2\WEB-INF`

- `move f:\temp\lib to C:\Temp\Apache-Software-Foundation\jakarta-tomcat-3.3.2\webapps\axis2\WEB-INF`

- `move f:\temp\repository to C:\Temp\Apache-Software-Foundation\jakarta-tomcat-3.3.2\webapps\axis2\WEB-INF`

- `move f:\temp\samples to C:\Temp\Apache-Software-Foundation\jakarta-tomcat-3.3.2\webapps\axis2\WEB-INF`

- `move f:\temp\repository to C:\Temp\Apache-Software-Foundation\jakarta-tomcat-3.3.2\webapps\axis2\WEB-INF`

- in `axis2.xml` allow hot deployment by changing

change

```
<parameter name="hotdeployment">false</parameter>
<parameter name="hotupdate">false</parameter>
```

to

```
<parameter name="hotdeployment">true</parameter>
<parameter name="hotupdate">true</parameter>
```

This change will enable Axis2 to detect if a new service (an *.arr file) is copied into the \services directory.

But if the service has a scope=application in its services.xml file, and the service has already been accessed, then the loaded web service class will not be replaced until tomcat is restarted or the axis2 context has been reloaded. Thus, changing and re-deploying the service would not result in the loaded class being terminated, even though the service has been re-deployed.

I have included later a sample ant task for reloading the tomcat \axis2.

7. Install SOAPMonitor

- copy "C:\Temp\Apache-Software-Foundation\apache-tomcat-5.5.25\webapps\axis2\WEB-INF\lib\soapmonitor-1.3.jar"

to

```
C:\Temp\Apache-Software-Foundation\apache-tomcat-5.5.25\webapps\axis2
```

- open cmd in this directory and run the jar which unpacks the soapmonitor.jar

```
> jar -xf soapmonitor-1.3.jar
```

- edit "C:\Temp\Apache-Software-Foundation\apache-tomcat-5.5.25\webapps\axis2\WEB-INF\web.xml"

- and add the soap monitor servlet definitions after the class AxisAdminServlet.

```
<servlet>
  <servlet-name>AxisAdminServlet</servlet-name>
  <display-name>Apache-Axis AxisAdmin Servlet (Web Admin)
</display-name>
  <servlet-class>
    org.apache.axis2.transport.http.AxisAdminServlet
  </servlet-class>
</servlet>
```

- add

```
<servlet>
  <servlet-name>SOAPMonitorService</servlet-name>
  <display-name>SOAPMonitorService</display-name>
  <servlet-class>
    org.apache.axis2.soapmonitor.servlet.SOAPMonitorService
  </servlet-class>
  <init-param>
    <param-name>SOAPMonitorPort</param-name>
    <param-value>5001</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>
```

```
<servlet-mapping>
  <servlet-name>SOAPMonitorService</servlet-name>
  <url-pattern>/SOAPMonitor</url-pattern>
</servlet-mapping>
```

- edit "C:\Temp\Apache-Software-Foundation\apache-tomcat-5.5.25\webapps\axis2\WEB-INF\conf\axis2.xml"

- add the reference for the soapmonitor after address reference

```
<!-- Comment this to disable Addressing -->
<module ref="addressing"/>

<module ref="soapmonitor"/>
```

8. Test SOAPMonitor

- Stop and restart tomcat then start soapmonitor in a browser

<http://localhost:8080/axis2/SOAPMonitor>

- and in another browser read the axis2 version and check the soap payload in the soapmonitor/

<http://localhost:8080/axis2/services/Version/getVersion>

9. Install tcpmon

- rename "C:\Temp\Apache-Software-Foundation\apache-tomcat-5.5.25\webapps\ROOT\favicon.icoxxxx"

to prevent the tcp monitor from reading the favicon.ico in the trace

- move f:\temp\tcpmon.jar
to

C:\Temp\Apache-Software-Foundation\apache-tomcat-5.5.25\webapps\axis2

- open cmd and start the tcp monitor

```
> java -jar tcpmon.jar
```

- set local port: 9080
server port: 8080

- and select Add Monitor, then open a browser to

<http://localhost:9080/axis2/services/Version/getVersion>

- and wait about 30 seconds for the monitor to display the following:

the request

```
GET /axis2/services/Version/getVersion HTTP/1.1
Host: localhost:9080
```

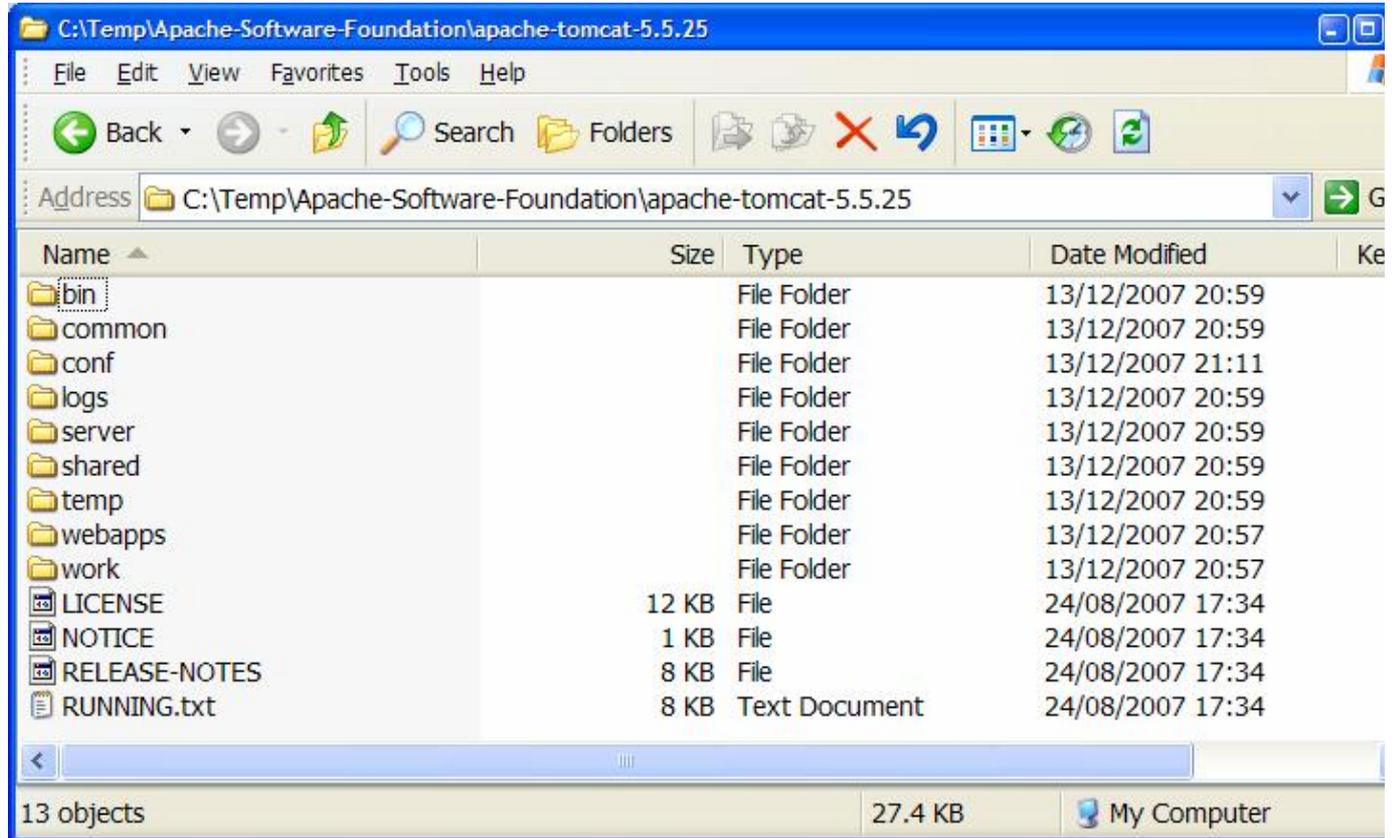
the response

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/xml; charset=UTF-8
Transfer-Encoding: chunked
```

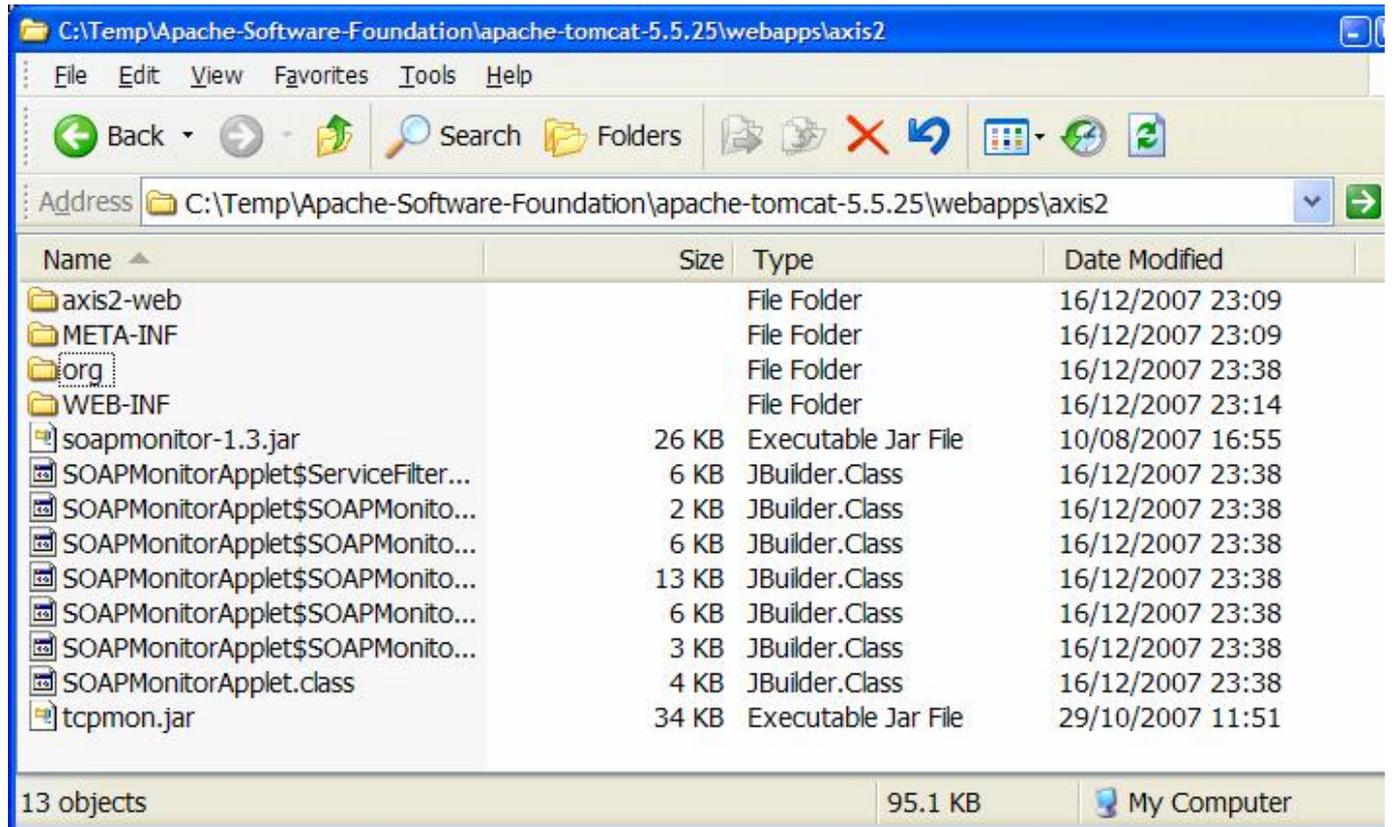
```
<ns:return>Hello I am Axis2 version service , My version is 1.3</ns:return>
```

10. Recheck the directory structure

Tomcat



Axis2



11. Forcing a reload from an ant task

Even with server side `axis2.xml` set for auto deployment, a reload of axis2 maybe be necessary to force an active web service class to be removed.

- add a manager account to `C:\Temp\Apache-Software-Foundation\apache-tomcat-5.5.25\conf\tomcat-users.xml`

```
<role rolename="manager" />
<user username="manager" password="axis2" roles="manager" />
```

- in an ant `build.xml` add the following task (ant reload)

```
<property name="tomcat.home" value="../../../../../../../" />

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

<taskdef name="reload"
classname="org.apache.catalina.ant.ReloadTask">
  <classpath refid="build.tomcat.path" />
</taskdef>

<target name="reload">
  <reload url=http://localhost:8080/manager
    username="manager"
    password="axis2"
    path="/axis2" />
</target>2
```

² <http://frotscher.com/axis2-book.html>

Using the Samples

With all the tools installed and the various files install, the samples will be demonstrated.

Since some of the samples have the same service name, I found the best results for testing was

1. Delete all the *.arr files in "...\axis2\WEB-INF\services\"
2. I deployed my service by coping the *.arr to \services, but the Axis2 Web admin servlet can also be used.
3. If there is still a problem with deployment of the new service then restart tomcat, because the scope may not be request and check the tomcat output.

(Recommended in most the Readme.txt)

pojo

- read README.txt

- open a cmd in C:\Temp\Apache-Software-Foundation\apache-tomcat-5.5.25\webapps\axis2\WEB-INF\samples\pojo

- > ant clean

- > ant // generate service

- copy AddressBookService.aar to \services

- > ant rpc.client

- > ant adb.client -Dwsdl=<http://localhost:8080/axis2/services/AddressBookService?wsdl>

Some improvements

- in AddressBookService.java add the following code to follow the deployment, class created and destroy.

```
import org.apache.axis2.context.ServiceContext;

public class AddressBookService
{
    static
    {
        System.out.println("+++++ AddressBookService:loaded ");
    }

    public AddressBookService()
    {
        System.out.println("***** AddressBookService()");
    }

    protected void finalize()
    {
        System.out.println("----- AddressBookService:finalize");
    }

    public void init(ServiceContext serviceContext)
    {
        System.out.println("===== AddressBookService:init");
    }

    public void destroy(ServiceContext serviceContext)
    {
        System.out.println("===== AddressBookService:destroy");
    }
    .....
}
```

- in the build.xml add

```
<project name="pojo" basedir="." default="generate.service">
...
<property name="tomcat.home" value="../../../../../../../" />

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

<taskdef name="reload" classname="org.apache.catalina.ant.ReloadTask">
  <classpath refid="build.tomcat.path" />
</taskdef>

<target name="reload">
  <reload url="http://localhost:8080/manager"
    username="manager"
    password="axis2"
    path="/axis2" />
</target>
</project>
```

- try the following reload

```
- > ant // generate changed service
- copy AddressBookService.aar to \services
- > ant reload
- > ant rpc.client
```

- try the following to debug of the client

- add the /classes to the client path, so that the logging files will be found

```
<path id="client.class.path">
  <fileset dir="${axis2.home}/lib">
    <include name="*.jar" />
  </fileset>
  <fileset dir="${dest.dir.lib}">
    <include name="*.jar" />
  </fileset>

  <pathelement location="${axis2.home}/classes"/>

</path>
```

- change the logging to output debug for the client

```
- edit C:\Temp\Apache-Software-Foundation\apache-tomcat-5.5.25\webapps\axis2\
WEB-INF\classes\log4j.properties
```

```
log4j.rootCategory=DEBUG, CONSOLE
#log4j.rootCategory=INFO, CONSOLE
```

- change the logging to output to axis2.log file for the client

```
- edit C:\Temp\Apache-Software-Foundation\apache-tomcat-5.5.25\webapps\axis2\
WEB-INF\classes\log4j.properties
```

log4j.rootCategory=DEBUG, LOGFILE

```
"C:\Temp\Apache-Software-Foundation\apache-tomcat-5.5.25\webapps\axis2\
WEB-INF\samples\pojo\axis2.log"
```

databinding

- read README.txt

```
>ant download.jars
```

```
>ant
```

```
>ant run.client -Durl=http://localhost:8080/axis2/services/StockQuoteService -Dsymbol=IBM
```

faulthandling

- see Readme.txt

mtom

- see Readme.txt

- The axis2.xml parameters can be changed. See the manual

```
<parameter name="enableMTOM">false</parameter>
to
<parameter name="enableMTOM">true</parameter>
```

- The test client is as follows:

```
ant run.client -Dfile build.xml -Ddest test.xml
```

with the build.xml file being copied to

```
C:\apache-tomcat-5.5.25\bin\test.xml
```

pojoguide

- see Readme.txt

pojoguidespring

- todo

quickstart

- see Readme.txt & restart tomcat

quickstartadb

- see Readme.txt & restart tomcat

quickstartaxiom

- see Readme.txt

quickstartjibx

- see Readme.txt

quickstartxmlbeans

- see Readme.txt

servicelifecycle

- see Readme.txt

Sample input for client

```
>ant run.client
```

```
[java] =====Welcome to Libraray client=====
[java]
[java] To list All books type 1
[java] To list Available books type 2
[java] To list Lend books type 3
[java] To register 4 - -userName -passWord
[java] To login 5 - -userName -passWord
[java] To lend a book 6 - -isbn -userName
[java] To return a book 7 - -isbn
[java] To exit type -1
[java]
[java]
[java] Enter service epr address :
```

```
http://localhost:8080/axis2/services/Library
```

```
[java] Type your command here :
```

```
1
```

```
[java] Isbn : 0-937175-77-3
[java] Author : John Bloomer
[java] Title : Power Programming with RPC
```

```
[java] Type your command here :
```

```
4 -userName xxx -passWord yyy
```

```
[java] Type your command here :
```

```
5 -userName xxx -passWord yyy
```

```
[java] Type your command here :
```

```
6 -isbn 0-937175-77-3 -userName xxx
```

```
[java] Title : Power Programming with RPC
[java] Isbn : 0-937175-77-3
[java] Author : John Bloomer
[java] Type your command here :
```

```
3
```

```
[java] Isbn : 0-937175-77-3
[java] Author : John Bloomer
[java] Title : Power Programming with RPC
[java]
[java] Type your command here :
```

```
7 -isbn 0-937175-77-3
```

```
[java] Type your command here :
```

```
3
```

```
[java] Type your command here :
```

```
-1
```

servicelifecycle

- see Readme.txt

soapwithattachments

- read README.txt

- The test client is as follows:

```
ant run.client -Dfile build.xml -Ddest junk.xml
```

with the build.xml file being copied to

```
C:\Temp\Apache-Software-Foundation\apache-tomcat-5.5.25\bin\junk.xml
```

version

- see Readme.txt

- delete the original version web service before deploying the new service as there will be a duplicate web service name

<http://localhost:8080/axis2/services/Version/getVersion>

yahoojsonsearch

- see Readme.txt

yahoorestsearch

- see Readme.txt

userguide

- see Readme.txt and the manual.

This sample uses both a client and a server `axis2.xml` file. The Web service `axis2.xml` file is found in the tomcat `axis2` context and used by the Web Service. While the client `axis2.xml` file (found in the this sample `\conf`) is input as an argument to the client Java VM. For the nonblocking samples, the newer `AxisCallback` class be used instead of the older `Callback` class (see the java docs).

This caused me many problems until I understand that the client could have its own `axis2.xml` and `\repository`.

- Build the logging module (sample-logging) and copy to both `\modules` for the web service and `\repository\modules` for the client.

- The sample logging handler must be added to the both `axis2.xml` files - the one used for startup of tomcat services and the one used by the client.

- For startup change:

```
C:\Temp\Apache-Software-Foundation\apache-tomcat-5.5.25\webapps\axis2\WEB-INF\conf\axis2.xml
```

- For client change:

```
C:\Temp\Apache-Software-Foundation\apache-tomcat-5.5.25\webapps\axis2\
WEB-INF\samples\userguide\conf\axis2.xml
```

- by adding to the 2 files in the four phases (See the manual)

```
<phase name="loggingPhase" />
```

- add to `build.xml`

```
<target name="run.client.all"
  depends="run.client.ping,
    run.client.blocking,
    run.client.blockingdual,
    run.client.nonblocking,
    run.client.nonblockingdual,
    run.client.servicewithmodule,
    run.client.rest,
    run.client.tcp,
    run.client.mail,
    run.client.wsa">
</target>

<target name="run.client.rest" depends="compile">
  <java classname="userguide.clients.RESTClient"
    classpathref="axis.classpath" fork="true">
    <jvmarg value="-Daxis2.repo=${mainDir}/repository"/>
    <jvmarg value="-Daxis2.xml=conf/axis2.xml"/>
  </java>
</target>

<target name="run.client.wsa" depends="compile">
  <java classname="userguide.clients.EchoBlockingWsaBasedClient"
    classpathref="axis.classpath" fork="true">
    <jvmarg value="-Daxis2.repo=${mainDir}/repository"/>
    <jvmarg value="-Daxis2.xml=conf/axis2.xml"/>
  </java>
</target>
```

- test by `ant run.client.all`

- the tcp client requirement more changes.

- add to build.xml

```
<target name="run.client.tcp" depends="compile">
  <java classname="userguide.clients.TCPClient"
    classpathref="axis.classpath" fork="true">
    <jvmarg value="-Daxis2.repo=${mainDir}/repository"/>
    <jvmarg value="-Daxis2.xml=conf/axis2.xml"/>
  </java>
</target>
```

- uncomment in the TCP transport receiver tomcat server axis2.xml

```
<!--Uncomment if you want to have TCP transport support-->
```

```
<transportReceiver name="tcp"
  class="org.apache.axis2.transport.tcp.TCPServer">
  <parameter name="port">7060</parameter>

  <!--If you want to give your own host address for EPR generation-->
  <!--uncomment the following paramter , and set it as you required.-->
  <!--<parameter name="hostname">tcp://myApp.com/ws</parameter>-->

</transportReceiver>
```

- change the port for the client from 8080 to 7060 in TCPClient.java

```
public class TCPClient

  private static String toEpr = "tcp://localhost:8080/axis2/services/MyService";
to
  private static String toEpr = "tcp://localhost:7060/axis2/services/MyService";

  OMNamespace omNs = fac.createOMNamespace(
    "tcp://localhost:8080/axis2/services/MyService", "example1");
to
  OMNamespace omNs = fac.createOMNamespace(
    "tcp://localhost:7060/axis2/services/MyService", "example1");
```

-The mail client demonstrates the use of the Robust In-Only MEP and is not a smtp/pop3 client.

- add to build.xml

```
<target name="run.client.mail" depends="compile">
  <java classname="userguide.clients.MailClient"
    classpathref="axis.classpath" fork="true">
    <jvmarg value="-Daxis2.repo=${mainDir}/repository"/>
    <jvmarg value="-Daxis2.xml=conf/axis2.xml"/>
  </java>
</target>
```

- The client needs to be changed

```
public class MailClient {
change
  private static String toEpr = "http://localhost:8080/axis2/services/myService";
to
  private static String toEpr = "http://localhost:8080/axis2/services/MyService";
```

jaxws-calculator

This sample has a bug and requires the downloading of updated war file and patching the maven repository file in order that the maven install executes correctly. There was initially no Readme, which was later added to the nightly build samples.

- download the following from the nightly build (<http://people.apache.org/dist/axis2/nightly/>)

1. distribution-SNAPSHOT-war.zip
2. distribution-SNAPSHOT-docs.zip
3. distribution-SNAPSHOT-bin.zip

- stop tomcat and remove the \axis2 context from tomcat (do not just rename)

- unzip the war, bin and docs

- copy the new axis2.war to tomcat

- start tomcat which installs the snapshot \axis2 context

- validate the new axis2, version should be SNAPSHOT

- copy the new snapshot jaws-calculator sample to the context where the old samples were copied

-the pom.xml file must be changed as follows:

```
<project xmlns="http://maven.apache.org/POM/4.0.0" instance="
  <groupId>org.apache.axis2</groupId>
    <artifactId>axis2-aar-maven-plugin</artifactId>
      <version>SNAPSHOT</version>
```

to

```
<version>1.3</version>
```

The build the web service using

```
> mvn clean install
```

```
[INFO] Scanning for projects...
```

```
[INFO] -----
```

```
[INFO] Building JAXWS Calculator Service
```

```
[INFO] task-segment: [clean, install]
```

This will take some time as maven will download many files into

```
"C:\Documents and Settings\<LoginName>\.m2\repository\...
```

before the build can be executed.

Then deploy the web service (jaxws-calculator-SNAPSHOT.aar) and follow the Readme.

Test by using the browser

<http://localhost:8080/axis2/services/CalculatorService/add?value1=1&value2=34>

Missing Samples

Services (in AXIS2_HOME/samples/userguide):

groovyService.aar: Demonstrates how to use Groovy classes in the processing of SOAP messages.

In AXIS2_HOME/samples Directory:

googleSpellcheck: Demonstrates both synchronous and non-synchronous use of the Google Web Service in a GUI.

The following batch file is saved in the \samples directory and used to initialize the path other environment variables needed for building the samples and developing new web services.

env.bat

```
rem TOMCAT

set CATALINA_BASE=c:\Temp\Apache-Software-Foundation\apache-tomcat-5.5.25
set CATALINA_HOME=c:\Temp\Apache-Software-Foundation\apache-tomcat-5.5.25

set PATH=c:\Temp\Apache-Software-Foundation\apache-tomcat-5.5.25\bin;

rem AXIS2

set AXIS2_HOME=C:\Temp\Apache-Software-Foundation\apache-tomcat-5.5.25\webapps\axis2\WEB-INF
set AXIS2_LIB=%AXIS2_HOME%\lib

rem JAVA

rem set JAVA_HOME=C:\Program Files\Java\jdk1.5.0_09

rem ANT

set ANT_HOME=C:\Temp\Apache-Software-Foundation\apache-ant-1.7.0

rem PATH

set PATH=C:\WINDOWS\system32;%PATH%

rem ant
set PATH=%PATH%;C:\Temp\Apache-Software-Foundation\apache-ant-1.7.0\bin

rem maven
set PATH=%PATH%;C:\Temp\Apache-Software-Foundation\apache-maven-2.0.8\bin

rem java
set PATH=%PATH%;C:\Program Files\Java\jdk1.5.0_09\bin

rem axis2
set PATH=%PATH%;C:\Temp\Apache-Software-Foundation\apache-tomcat-5.5.25\webapps\axis2\WEB-INF\bin

rem CLASSPATH

set CLASSPATH=C:\Program Files\Java\jdk1.5.0_09

set CLASSPATH=%CLASSPATH%;C:\Temp\Apache-Software-Foundation\apache-tomcat-5.5.25\common\lib\servlet-api.jar

set AXISCLASSPATH=%AXIS2_LIB%\commons-discovery-0.2.jar

set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\activation-1.1.jar;
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\annogen-0.1.0.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\axiom-api-1.2.5.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\axiom-dom-1.2.5.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\axiom-impl-1.2.5.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\axis2-adb-1.3.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\axis2-adb-codegen-1.3.jar
```

```

set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\axis2-clustering-1.3.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\axis2-codegen-1.3.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\axis2-fastinfoset-1.3.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\axis2-java2wsdl-1.3.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\axis2-jaxbri-1.3.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\axis2-jaxws-1.3.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\axis2-jaxws-api-1.3.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\axis2-jibx-1.3.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\axis2-json-1.3.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\axis2-jws-api-1.3.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\axis2-kernel-1.3.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\axis2-metadata-1.3.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\axis2-mtompolicy-1.3.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\axis2-saa-j-1.3.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\axis2-saa-j-api-1.3.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\axis2-spring-1.3.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\axis2-xmlbeans-1.3.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\backport-util-concurrent-2.2.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\commons-codec-1.3.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\commons-fileupload-1.1.1.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\commons-httpclient-3.0.1.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\commons-io-1.2.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\commons-logging-1.1.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\geronimo-annotation_1.0_spec-1.1.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\geronimo-jms_1.1_spec-1.1.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\httpcore-4.0-alpha5.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\httpcore-nio-4.0-alpha5.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\httpcore-nio-ssl-4.0-alpha5.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\jaxb-api-2.0.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\jaxb-impl-2.0.5.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\jaxb-xjc-2.0.5.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\jaxen-1.1.1.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\jettison-1.0-RC1.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\jibx-bind-1.1.5.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\jibx-run-1.1.5.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\juli-6.0.10.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\log4j-1.2.14.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\mail-1.4.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\mex-impl-1.3.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\neethi-2.0.2.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\soapmonitor-1.3.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\stax-api-1.0.1.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\tribes-6.0.10.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\woden-1.0-incubating-M7b.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\wsdl4j-1.6.2.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\wstx-asl-3.2.1.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\xalan-2.7.0.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\xbean-2.2.0.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\xercesImpl-2.8.1.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\xml-apis-1.3.03.jar
set AXISCLASSPATH=%AXISCLASSPATH%;%AXIS2_LIB%\XmlSchema-1.3.2.jar

set CLASSPATH=%CLASSPATH%;%AXISCLASSPATH%

```