

Creating & Deploying AXIS2 WebService using ANTS Build

[White Paper]

Nandakumar Kuthalaraja
392330



Confidentiality Statement

© 2011 TATA CONSULTANCY SERVICES LIMITED
All rights reserved.

This document contains confidential information of TATA CONSULTANCY SERVICES LIMITED, which is provided for the sole purpose of permitting the recipient to evaluate any proposal submitted herewith or separately. In consideration of receipt of this document, the recipient agrees to maintain such information in confidence and to not reproduce or otherwise disclose this information to any person outside the group directly responsible for evaluation of its contents, except that there is no obligation to maintain the confidentiality of any information which was known to the recipient prior to receipt of such information from TATA CONSULTANCY SERVICES LIMITED, or becomes publicly known through no fault of recipient, or is received without obligation of confidentiality from a third party owing no obligation of confidentiality to TATA CONSULTANCY SERVICES LIMITED.

This document is the property of TATA CONSULTANCY SERVICES LIMITED and TATA CONSULTANCY SERVICES LIMITED may require the return of this document at any time at their sole discretion. Unauthorized access or copying is prohibited without the prior written permission of TATA CONSULTANCY SERVICES LIMITED.

All product and company names mentioned herein may be trademarks of their respective owners.

ABSTRACT

This Whitepaper will provide Web Services concepts and step by step process to develop and deploy a web service using ANTS build. The working example will be very much useful and easy to understand for the beginners to develop & deploy a web services in java.

ABOUT THE AUTHOR

Having 01 plus years of working experience in java/j2ee. Poses 4 months of working experience in developing the web services in java. Currently working for Century Link Corporation. Developed an Internal System to collect client billing information from associates using J2EE.

CONTENTS

1. What is a Web Services?
2. Why Web Services?
3. What is SOAP?
4. What is WSDL?
5. Steps for Developing & Hosting a Web Service.
6. Running Web Service

1. What is a Web Services?

A Web services is a Web-based enterprise application, accessible on the web through a URL, which is accessed by clients using, XML-based protocol, such as SOAP and accepted Internet transport protocols, such as HTTP to exchange data.

2. Why Web Services?

Many types of information available across the globe with different data formats, different platforms and different networks to communicate with those different data we required application-to-application integration over the Internet. In order to address the current challenges of distributed computing and B2B integration, software industry has been rapidly evolving XML-based Web services technologies as the approach to solving these problems.

3. What is SOAP?

SOAP, which stands for Simple Object Access Protocol, defines the Request and the Response message format in a Web Service Application. Simply put, it is nothing but a XML structure adhering to a well defined Schema. A Web Service client sends Request to a Web Service in the form of SOAP message and gets the response back in SOAP message only. Let us look into the structure of the SOAP message in the following section.

A SOAP message consists of a SOAP Envelope which forms the outer most layers and it merely serves as a container for the SOAP Header and SOAP Body Elements. The SOAP Header will carry information that doesn't constitute to the original part of the message. For example, a SOAP Header may carry some Application Specific Information language such the Locale information.

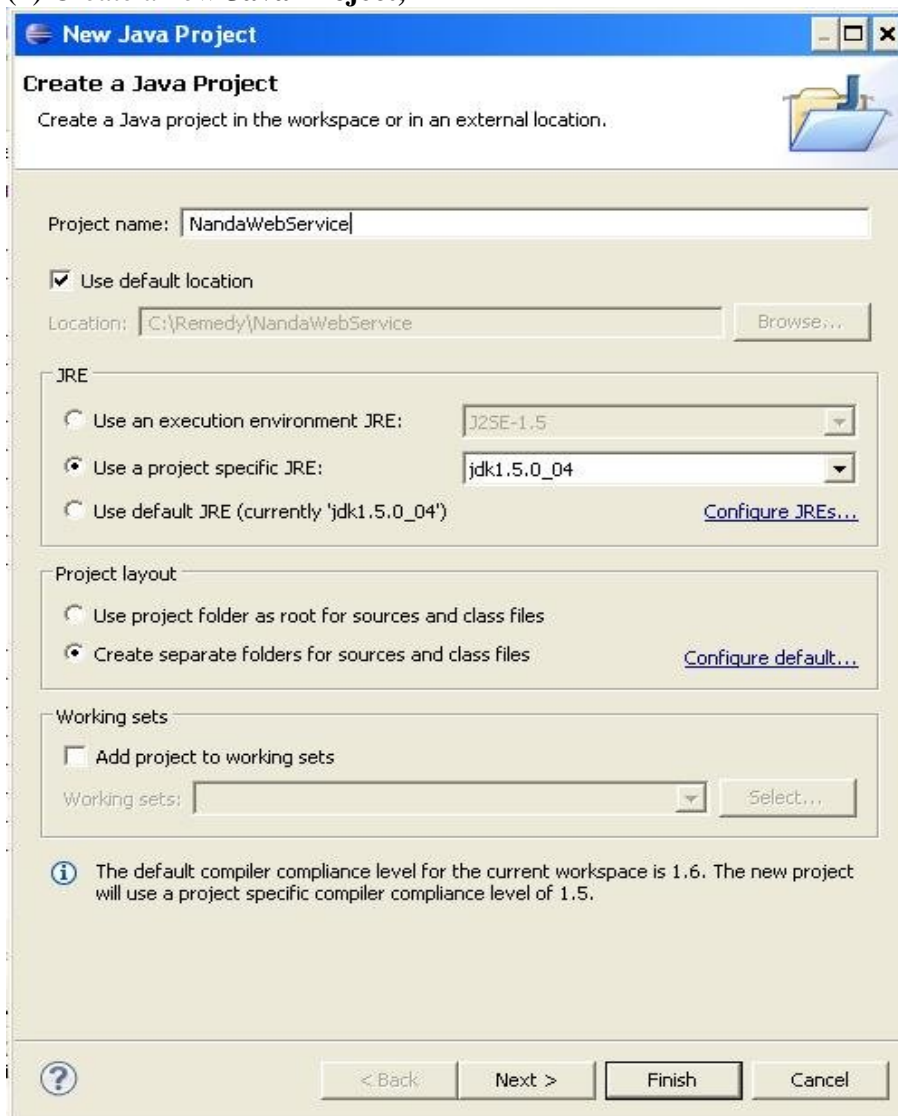
This SOAP Header is optional only. The SOAP Body consist of the original data that needs to be processed by the Web Service Application. Consider the following xml snippet which illustrates the SOAP message.

4. What is WSDL?

Web Services Description Language (WSDL) is again a XML Document that describes the definition of Web Services so that they can be accessible by the Client. Technically, **WSDL Document** contains the name of the Web Services, the arguments it is going to accept, the URL through which it can be accessed.

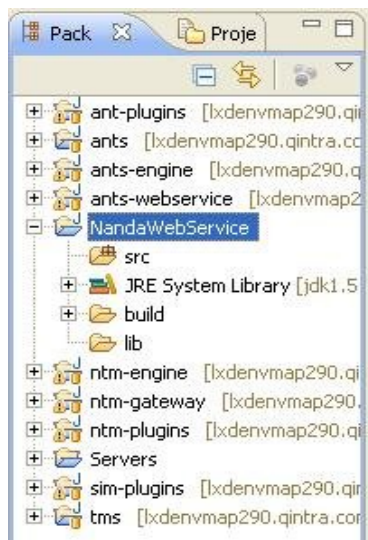
5. Steps for Developing & Hosting a Web Service (as Standalone Java Project)

(1) Create a new **Java Project**,

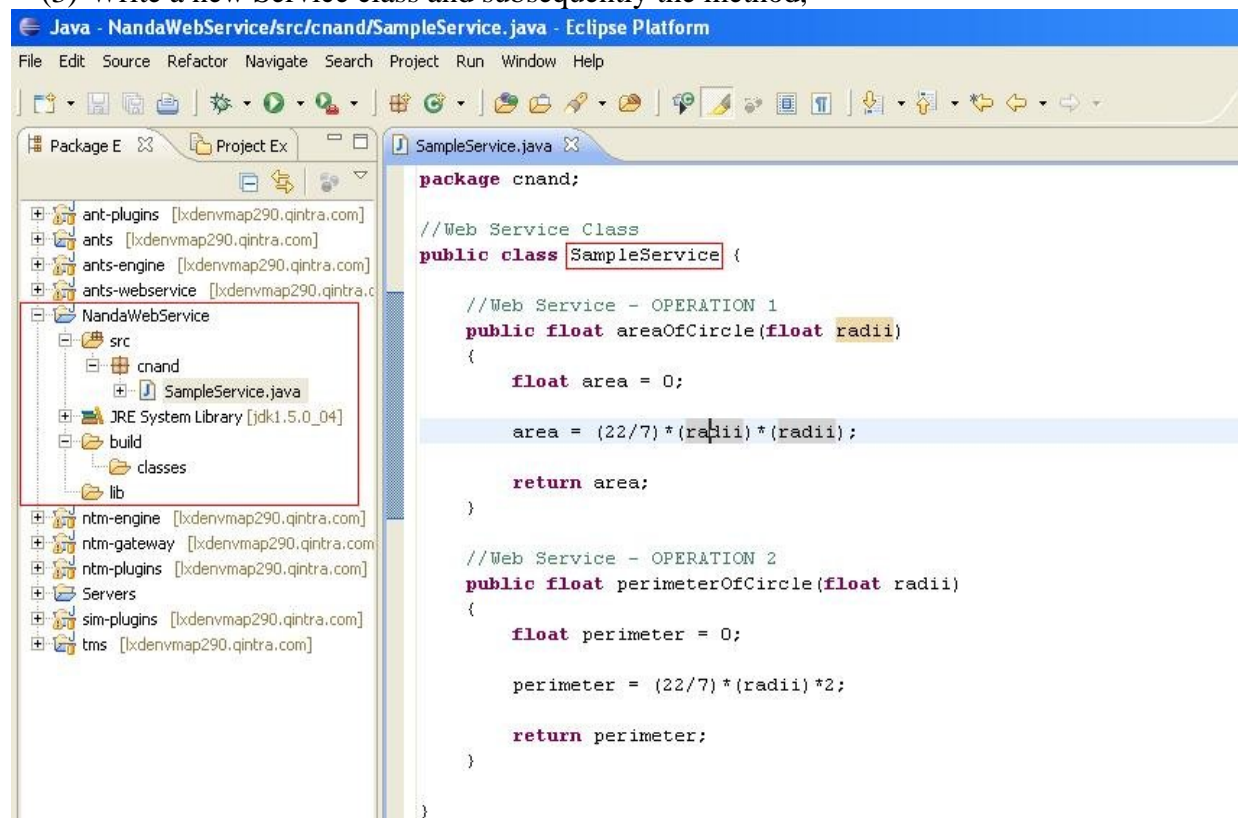


Creating & Deploying AXIS2 WebService using ANTS Build

(2) Add new folders – Build & lib

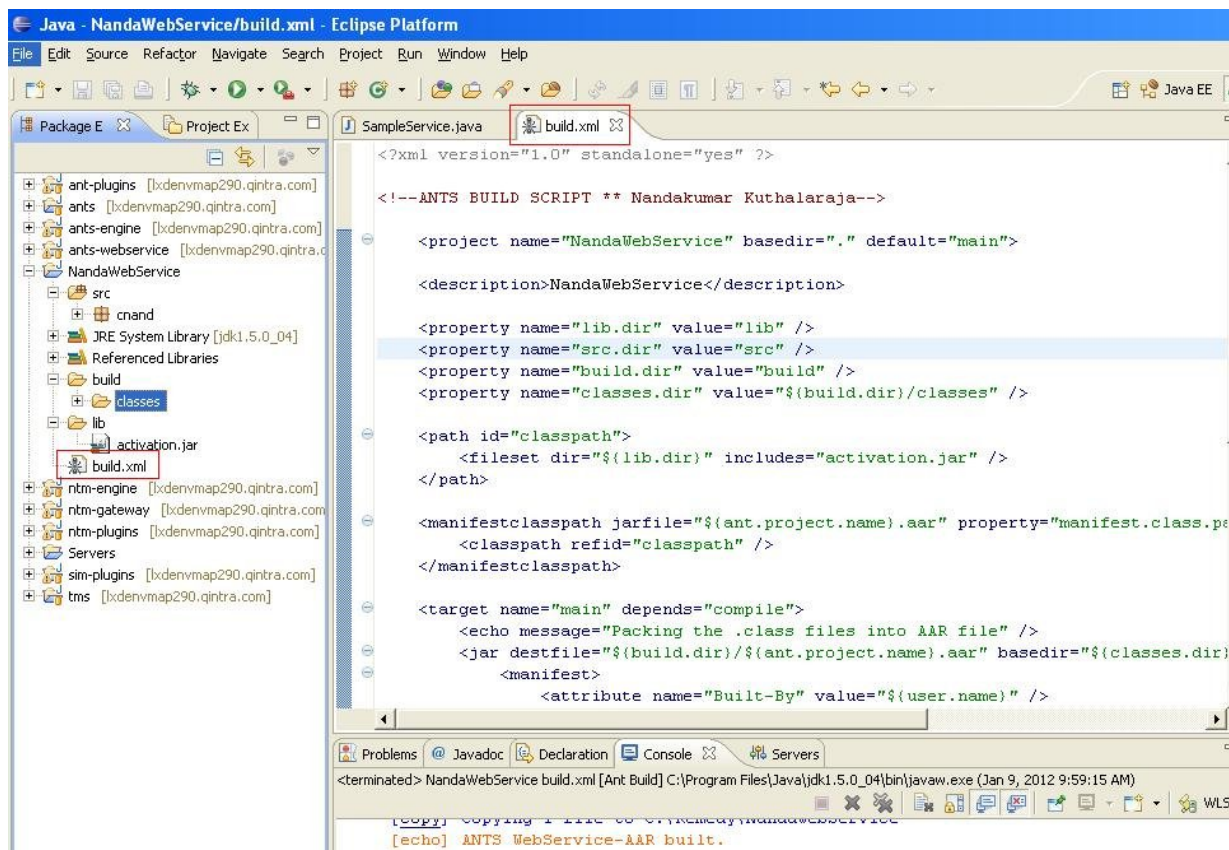


(3) Write a new Service class and subsequently the method,



Creating & Deploying AXIS2 WebService using ANTS Build

(4) Writing build.xml for the Web Service (ANTS build Script)



This Build script will perform compilation and create AAR (AXIS ARCHIVE FILE) for deploying the WebService in the tomcat server.

(5) build.xml

```
<?xml version="1.0" standalone="yes" ?>
<!--ANTS BUILD SCRIPT ** Nandakumar Kuthalaraja-->
  <project name="NandaWebService" basedir="." default="main">
    <description>NandaWebService</description>
    <property name="lib.dir" value="lib" />
    <property name="src.dir" value="src" />
    <property name="build.dir" value="build" />
    <property name="classes.dir" value="${build.dir}/classes" />
    <path id="classpath">
      <fileset dir="${lib.dir}" includes="activation.jar" />
    </path>
    <manifestclasspath jarfile="${ant.project.name}.aar" property="manifest.class.path">
      <classpath refid="classpath" />
    </manifestclasspath>
```

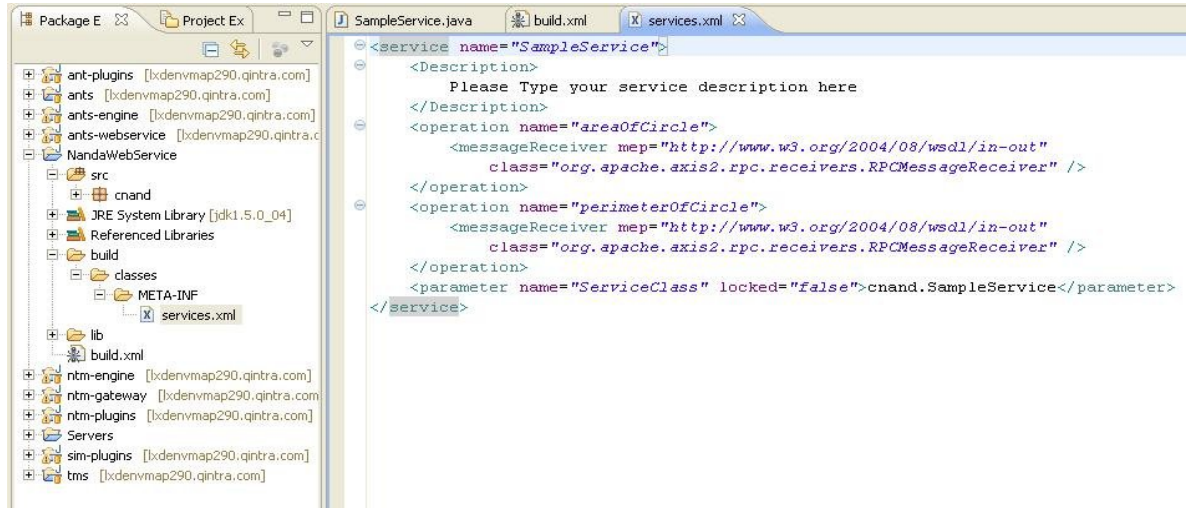
Creating & Deploying AXIS2 WebService using ANTS Build


```

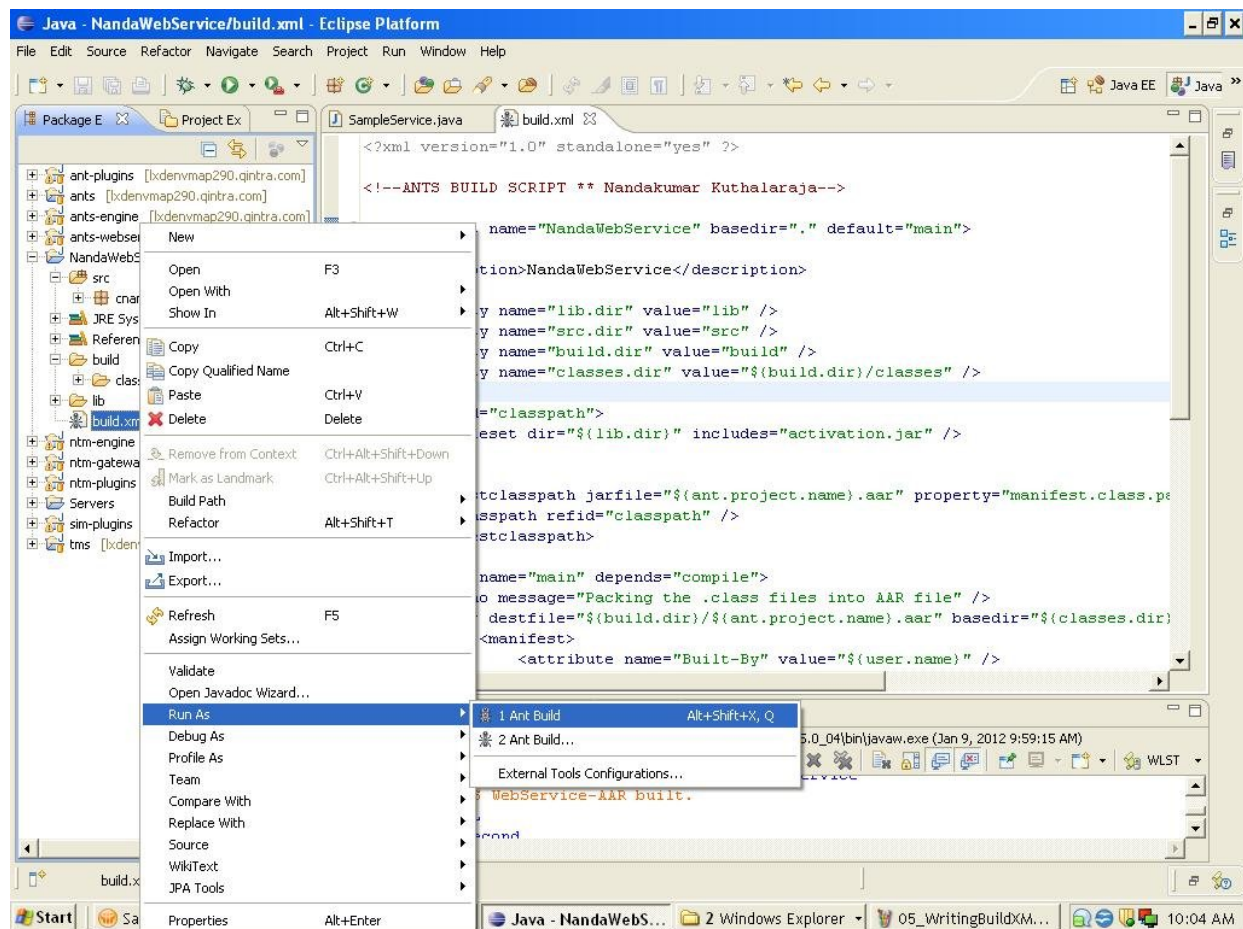
<target name="main" depends="compile">
    <echo message="Packing the .class files into AAR file" />
    <jar destfile="${build.dir}/${ant.project.name}.aar" basedir="${classes.dir}">
        <manifest>
            <attribute name="Built-By" value="${user.name}" />
            <attribute name="Class-Path" value="${manifest.class.path}" />
        </manifest>
    </jar>
    <copy file="${build.dir}/${ant.project.name}.aar" todir="." />
    <echo message="ANTS WebService-AAR built." />
</target>
<!-- Clear up intermediate dirs -->
<target name="clean">
    <echo message="Classes and build directories ..." />
    <delete dir="${build.dir}" />
    <mkdir dir="${build.dir}" />
    <mkdir dir="${classes.dir}" />
</target>
<!-- Generate classfiles files from source code in src -->
<target name="compile">
    <echo message="Compiling the java source files..." />
    <javac srcdir="${src.dir}" destdir="${classes.dir}" classpathref="classpath"
debug="on" debuglevel="lines,vars,source">
        <exclude name="com/qwest/remedy/NandaWebService/test/**" />
    </javac>
</target>
</project>

```

(6) Writing Service.xml for the Web Service

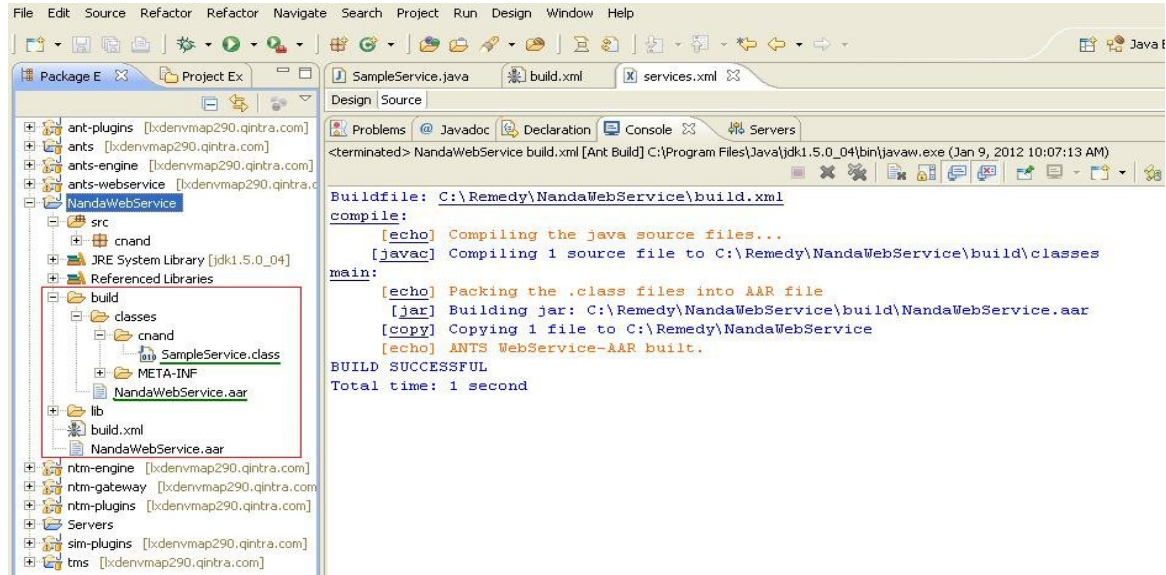


(7) Compiling the Java Project & Generating AAR file Using ANTS Build Script (build.xml)

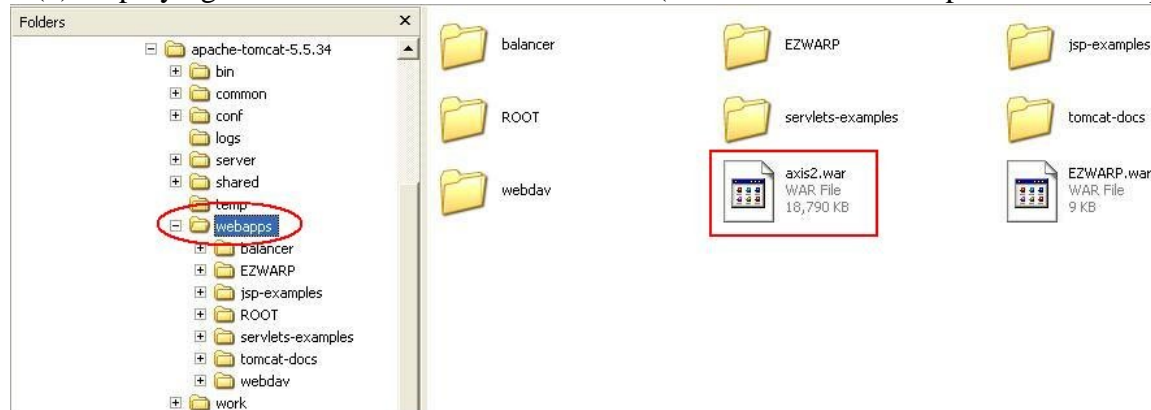


(8) Console window after the built

Creating & Deploying AXIS2 WebService using ANTS Build



(9) Deploying AXIS Runtime in Tomcat Server (Get axis2.war from Apache axis web portal)

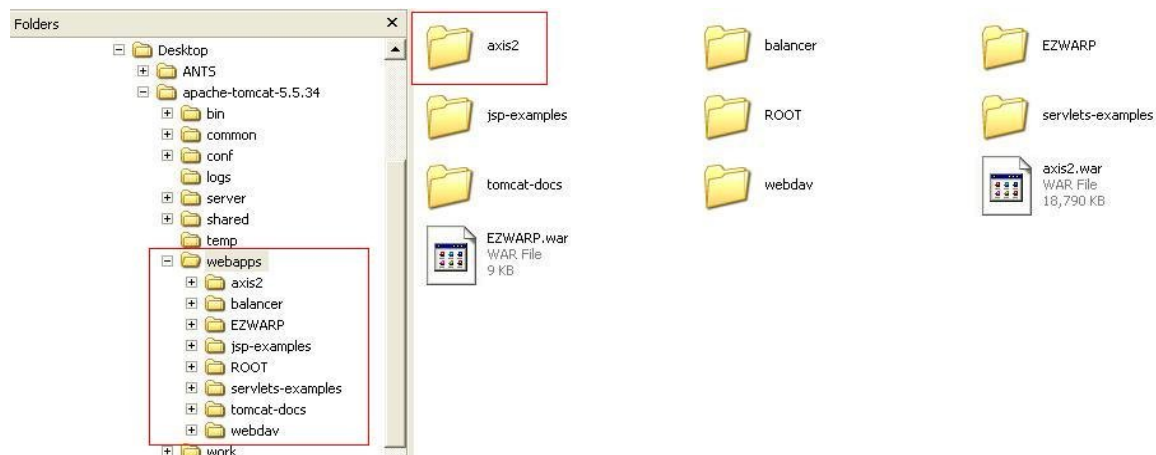


(10) Restart the Tomcat Server the Console of Tomcat will look like

Creating & Deploying AXIS2 WebService using ANTS Build

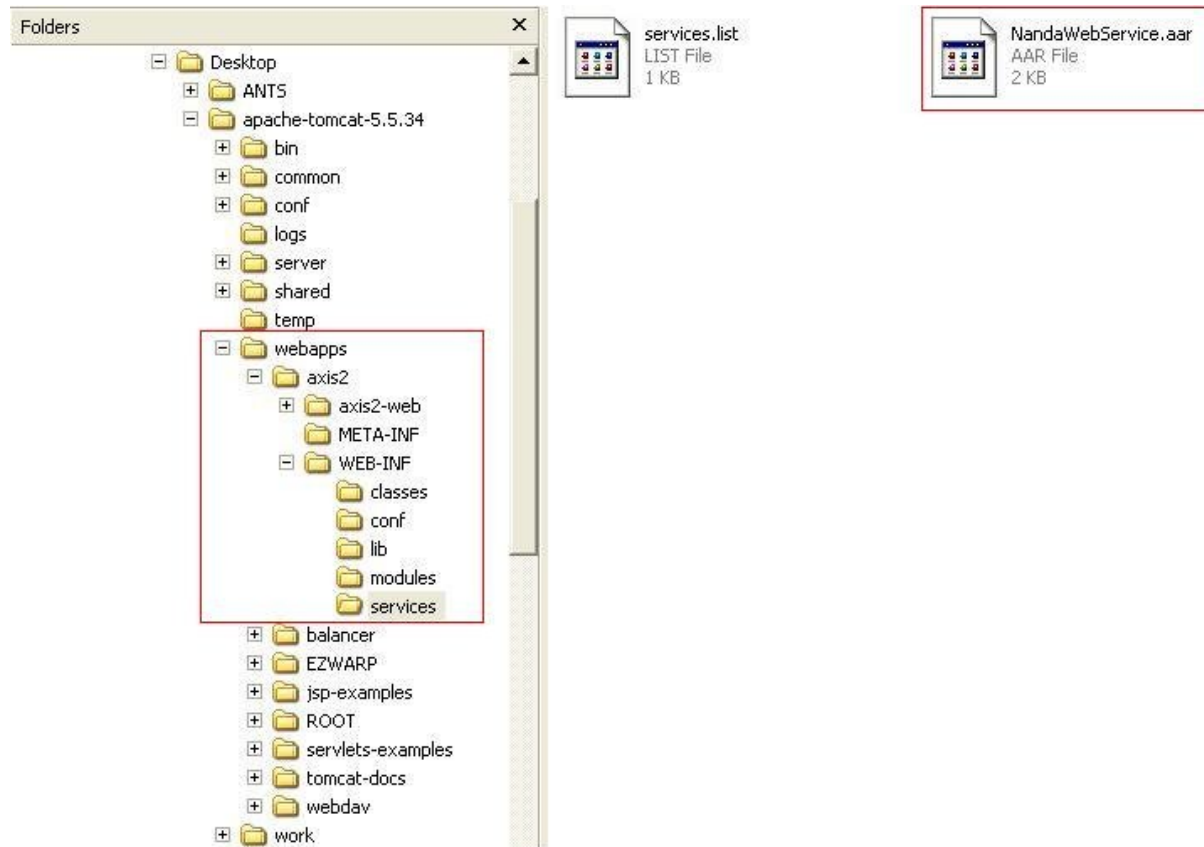

```
Tomcat
Jan 9, 2012 10:13:02 AM org.apache.catalina.core.AprLifecycleListener init
INFO: Loaded APR based Apache Tomcat Native library 1.1.22.
Jan 9, 2012 10:13:02 AM org.apache.catalina.core.AprLifecycleListener init
INFO: APR capabilities: IPv6 [false], sendfile [true], accept filters [false], random [true].
Jan 9, 2012 10:13:02 AM org.apache.coyote.http11.Http11AprProtocol init
INFO: Initializing Coyote HTTP/1.1 on http-8080
Jan 9, 2012 10:13:02 AM org.apache.coyote.ajp.AjpAprProtocol init
INFO: Initializing Coyote AJP/1.3 on ajp-8009
Jan 9, 2012 10:13:02 AM org.apache.catalina.startup.Catalina load
INFO: Initialization processed in 1297 ms
Jan 9, 2012 10:13:03 AM org.apache.catalina.core.StandardService start
INFO: Starting service Catalina
Jan 9, 2012 10:13:03 AM org.apache.catalina.core.StandardEngine start
INFO: Starting Servlet Engine: Apache Tomcat/5.5.34
Jan 9, 2012 10:13:03 AM org.apache.catalina.core.StandardHost start
INFO: XML validation disabled
Jan 9, 2012 10:13:03 AM org.apache.catalina.startup.HostConfig deployWAR
INFO: Deploying web application archive axis2.war
[INFO] Deploying module: addressing - file:/C:/Documents and Settings/392330/Desktop/apache-tomcat-5.5.34/webapps/axis2/WEB-INF/modules/addressing-1.4.1.mar
[INFO] Deploying module: script-1.41 - file:/C:/Documents and Settings/392330/Desktop/apache-tomcat-5.5.34/webapps/axis2/WEB-INF/modules/axis2-scripting-1.41.mar
[INFO] Deploying module: metadataExchange-1.41 - file:/C:/Documents and Settings/392330/Desktop/apache-tomcat-5.5.34/webapps/axis2/WEB-INF/modules/mex-1.41.mar
[INFO] Deploying module: ping-1.41 - file:/C:/Documents and Settings/392330/Desktop/apache-tomcat-5.5.34/webapps/axis2/WEB-INF/modules/ping-1.41.mar
[INFO] Deploying module: soapmonitor-1.41 - file:/C:/Documents and Settings/392330/Desktop/apache-tomcat-5.5.34/webapps/axis2/WEB-INF/modules/soapmonitor-1.41.mar
[INFO] Deploying Web service: version-1.4.1.aar - file:/C:/Documents and Settings/392330/Desktop/apache-tomcat-5.5.34/webapps/axis2/WEB-INF/services/version-1.4.1.aar
Jan 9, 2012 10:13:09 AM org.apache.catalina.startup.HostConfig deployWAR
INFO: Deploying web application archive EZWARP.war
Jan 9, 2012 10:13:09 AM org.apache.coyote.http11.Http11AprProtocol start
INFO: Starting Coyote HTTP/1.1 on http-8080
Jan 9, 2012 10:13:09 AM org.apache.coyote.ajp.AjpAprProtocol start
INFO: Starting Coyote AJP/1.3 on ajp-8009
Jan 9, 2012 10:13:09 AM org.apache.catalina.storeconfig.StoreLoader load
INFO: Find registry server-registry.xml at classpath resource
Jan 9, 2012 10:13:09 AM org.apache.catalina.startup.Catalina start
INFO: Server startup in 6847 ms
```

(11) AXIS2 Runtime folder gets created,

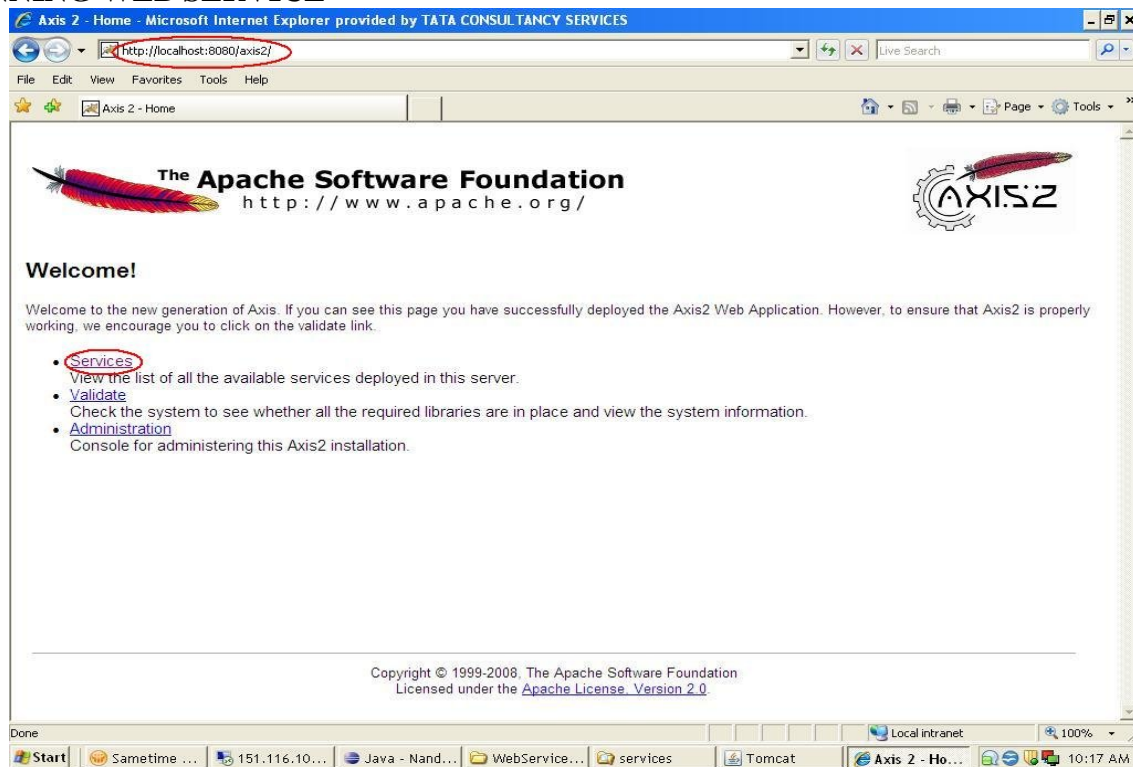


Creating & Deploying AXIS2 WebService using ANTS Build

(12) Deploy the AAR file built using the ANTS build into the,
\\...\\tomcat\\webapps\\axis2\\WEB-INF\\services

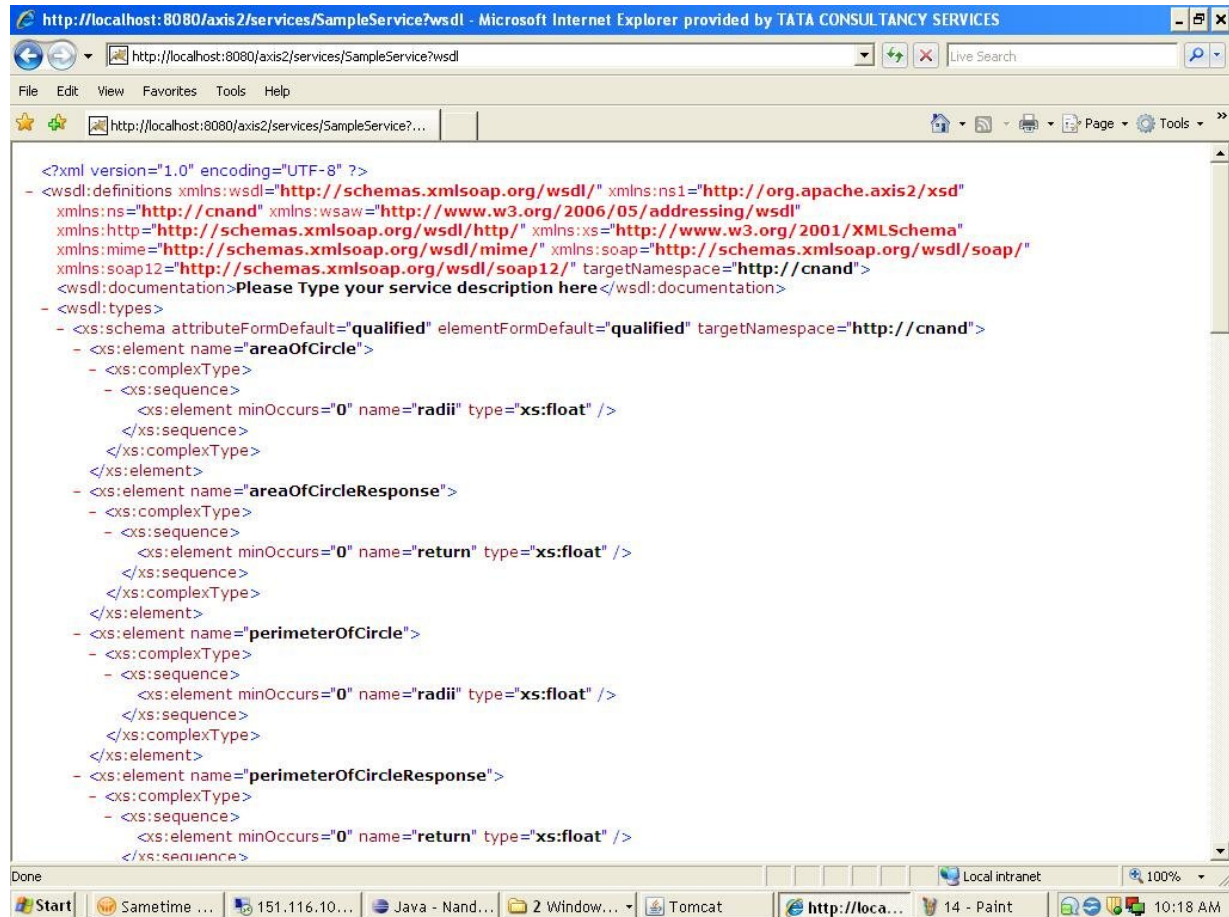


RUNNING WEB SERVICE



Creating & Deploying AXIS2 WebService using ANTS Build

WSDL



Now our WebService is ready to be consumed. Advantage of this methodology is the steps are common to deploy in servers running in any OS platform.

References

1. Java Web Services: Up and Running by Martin Kalin
2. Download URL : <http://ebooks.jsoft.ws/book.php?download&href=java-web-services-up-and-running.9780596521127.41654.chm>