


Apache Axis2: Next Generation Web Services

Asst. Prof. Dr. Kanda Runapongsa
(krunapon@kku.ac.th)
Department of Computer Engineering
Khon Kaen University


1



Agenda

- What and Why Apache Axis2?
- Understanding AXIOM
- Learning Axis2 Basics
- Understanding the Deployment Model
- Axis2 Installation
- Generating a Service and Deploying using POJO (Plain Old Java Object)
- Generating a Service and Deploying using AXIOM


2



What is Apache Axis2?


- Apache Axis2 is the core engine for Web services
- It is a complete re-design and re-write of widely used Apache Axis

3

 **What is Apache Axis?**


- Apache Axis is an implementation of the SOAP
 - An envelope that defines a framework for describing what is in a message and how to process it
 - A set of encoding rules for expressing instances of application-defined data types
 - A convention for representing remote procedure calls and responses.

4

 **Why Apache Axis2?**


- More flexible, efficient, and configurable in comparison to Axis1.x
- Supports SOAP 1.1 and SOAP 1.2 as well as REST style of Web services
- Support the easy addition of plug-in “modules”
 - WS-ReliableMessaging
 - WS-Coordination and WS-AtomicTransaction
 - WS-Security
 - WS-Addressing

5

 **Key Features of Apache Axis2 (1/4)**


- Speed: much faster than the old version
 - Use its own object model and StAX
- Low memory foot print
- AXIOM: extensible, highly performance, and developer convenient
 - Light-weight object model

6

 **Key Features of Apache Axis2 (2/4)**


- Hot deployment
 - New services can be added without having to shut down the server
- Asynchronous Web services
 - Supports asynchronous Web services invocation using non-blocking clients and transports
- MEPs support
 - Support Message Exchange Patterns (MEPs)

7

 **Key Features of Apache Axis2 (3/4)**


- Transport framework
 - A clean and simple abstraction for integrating and using Transports (over various protocols such as SMTP and FTP)
- WSDL support
 - Axis2 supports WSDL 1.1 and 2.0 which allows you to easily build stubs to access remote services

8

 **Key Features of Apache Axis2 (4/4)**


- Add-ons
 - Have been incorporated with WSS4J for security, Sandesha for reliable messaging, Kandula for coordination, atomic transaction, and business activity
- Compositions and Extensibility
 - Modules and phases improve support for composability and extensibility

9

 **Supported Specifications**


- SOAP 1.1 and 1.2
- Message Transmission Optimization Mechanism (MTOM), XML Optimized Packaging (XOP) and SOAP with Attachments
- WSDL 1.1, including both SOAP and HTTP bindings
- WS-Addressing
- WS-Policy
- SAAJ 1.1

10

 **Supported Transports and Data Bindings**


- Supported Transports
 - HyperText Transfer Protocol (HTTP)
 - Simple Mail Transfer Protocol (SMTP)
 - Java Message Service (JMS)
 - Transmission Control Protocol (TCP)
- Supported Data Bindings
 - Axis Data Binding (ADB)
 - XMLBeans
 - JibX
 - JaxMe

11

 **Tools Included in Axis2 version 1.1**


- Axis2 Web Application (Web App)
- WSDL2WS
 - Eclipse plug in / Maven2 WSDL2Code Plug in
- Service Archive Wizard
 - Eclipse plug in / Maven2 AAR Plug -in
- Java2WSDL
 - Maven 2 Java2WSDL Plug-in

12

 **Extension Modules**


- Apache Rampart: Supporting WS-Security (and soon WS-Policy)
- Apache Sandesha2: Supporting Ws-Reliable Messaging
- Apache Axis2 comes built in with a module that supports WS-Addressing

13

 **Agenda**


- What and Why Apache Axis2?
- Understanding AXIOM
- Learning Axis2 Basics
- Understanding the Deployment Model
- Axis2 Installation
- Generating a Service and Deploying using POJO (Plain Old Java Object)
- Generating a Service and Deploying using AXIOM

14

 **AXIOM (AXIS Object Model)**


- Objects are created “on demand” using a pull model
- Allows direct access to the underlying pull stream with or without building the tree
- Allows the event based navigation of the OM tree
- Support for storing binary data

15

 **Agenda**


- What and Why Apache Axis2?
- Understanding AXIOM
- Learning Axis2 Basics
- Understanding the Deployment Model
- Axis2 Installation
- Generating a Service and Deploying using POJO (Plain Old Java Object)
- Generating a Service and Deploying using AXIOM

16

 **Message Processing Stages**


- There are three main stages
 - Transport Receiver
 - Transport related processing
 - Dispatching
 - Finding service and operation
 - Message Receiver
 - Last handler of the chain

17

 **Contexts and Descriptions Hierarchy**


- Descriptors keep static information
 - Information extracted from deployment descriptors
- Contexts keep runtime information
- This information needs to be in various scope
- Good to keep contexts and descriptions separate

18

 **Agenda**


- What and Why Apache Axis2?
- Understanding AXIOM
- Learning Axis2 Basics
- Understanding the Deployment Model
- Axis2 Installation
- Generating a Service and Deploying using POJO (Plain Old Java Object)
- Generating a Service and Deploying using AXIOM

19

 **What's the Fuss with Deployment**

- Axis 1.x deployment requires you to
 - Modify the XML files
 - Call the admin client
 - Add to the classpath
 - Restart the server
- For a beginner, a bit of headache ☹

20

 **New Deployment Model**

- Archive based deployment
 - Bundle all together and drop in
- Directory based deployment (similar structure as archive)
- Hot deployment ☺
- An archive file can contain
 - Class files
 - Third party libraries
 - Any other resources required by the service

21

Axis2 Service

- Can be deployed as an archive (.aar) file or as a directory with all necessary resources
- Service configurations are given by the services.xml which contains
 - ServiceClass parameter
 - Namespaces
 - Expose transports
 - Operation
 - Modules to be engaged
 - Module configurations

22

Agenda

- What and Why Apache Axis2?
- Understanding AXIOM
- Learning Axis2 Basics
- Understanding the Deployment Model
- Axis2 Installation
- Generating a Service and Deploying using POJO (Plain Old Java Object)
- Generating a Service and Deploying using AXIOM

23

Axis2 Installation

- Download and install Java SDK
- Download and deploy a servlet container such as Apache Tomcat
- Download distribution from <http://ws.apache.org/axis2>
- Copy axis2.war to <TOMCAT>/webapps folder
- Start Tomcat server
- Start URL at <http://localhost:8080/axis2>

24

Download and Install Java SDK

- Download and install JavaSE SDK v1.5 update 9 which can be downloaded from
 - <http://gear.kku.ac.th/~krunapon/xmlws> >> tools >> JDK 1.5.0.09 with NetBeans Bundle
- Note that JavaSE 6 has compatibility problem with Apache Ant tool
- Set %JAVA_HOME% environment variable to the directory at where JavaSE SDK 1.5 is located

```

C:\WINDOWS\system32\cmd.exe
Microsoft windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\Documents and Settings\Krunapon>echo %JAVA_HOME%
C:\Program Files\Java\jdk1.5.0_09
    
```

Download a Servlet Container (Tomcat)

- In this case, we use Tomcat 5.5.20 which can be download from
 - <http://tomcat.apache.org/> >> Download >> Tomcat 5.x
 - <http://gear.kku.ac.th/~krunapon/xmlws> >> tools >> Tomcat 5.5.20
- Unzip apache-tomcat-5.5.20.zip

26

Deploy and Start Tomcat Server

- Set %CATALINA_HOME% environment variable to the Tomcat directory
- Start Tomcat server by running startup.bat which is at %CATALINA_HOME%/bin

```

C:\apache\tomcat\apache-tomcat-5.5.20\bin>startup.bat
Using CATALINA_BASE:   c:\apache\tomcat\apache-tomcat-5.5.20
Using CATALINA_HOME:   c:\apache\tomcat\apache-tomcat-5.5.20
Using CATALINA_TMPDIR: c:\apache\tomcat\apache-tomcat-5.5.20\temp
Using JRE_HOME:        C:\Program Files\Java\jdk1.5.0_09
C:\apache\tomcat\apache-tomcat-5.5.20\bin>_
    
```

27

Open the Homepage of Tomcat

<http://localhost:8080>

28

Download Apache Axis2 (1/2)

- Download Apache Axis2
- <http://ws.apache.org/axis2/download.cgi>
- Choose version 1.1.1

29

Download Apache Axis2 (2/2)

- Choose Standard Binary Distribution

- Extract axis2-1.1.1.zip
- Set %AXIS2_HOME% to the axis2 directory

30

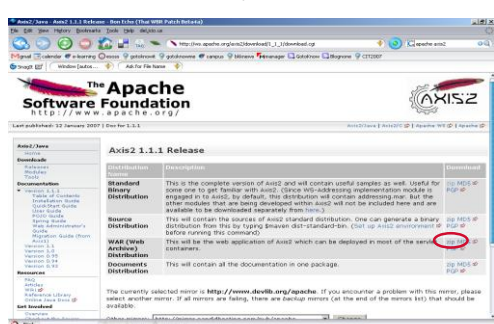
Install Apache Axis2 Overview

- Obtain axis2.war which can use one of these two methods
 - Download axis2.war from http://ws.apache.org/axis2/download/1_1_1/download.cgi
 - Create axis2.war by using Apache Ant which can be downloaded from <http://ant.apache.org/bindownload.cgi>
- Copy axis2.war to %CATALINA_HOME%/webapps directory

31

Download axis2.war

- In Distribution Name column, choose WAR (Web Archive Distribution)
- In Download column, choose zip

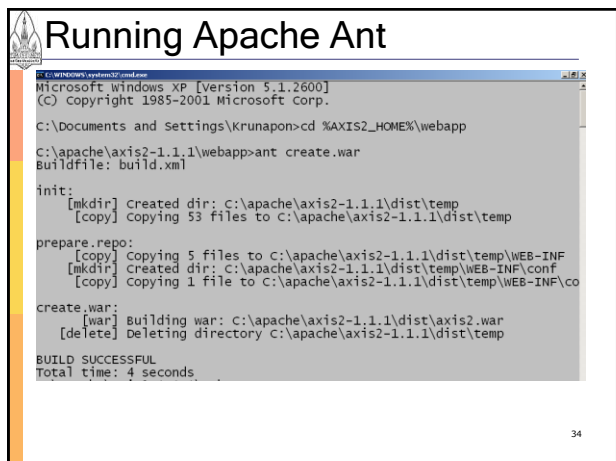


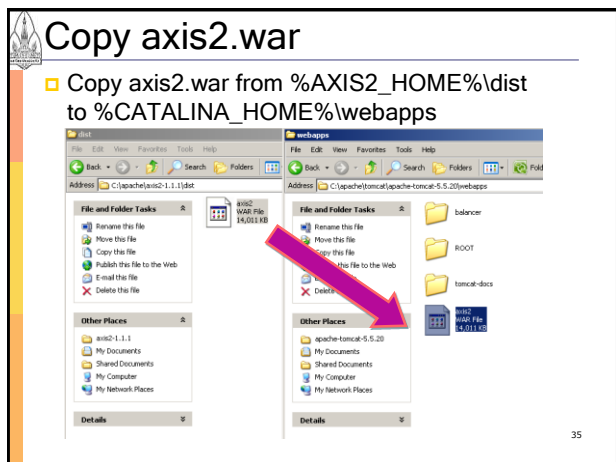
32

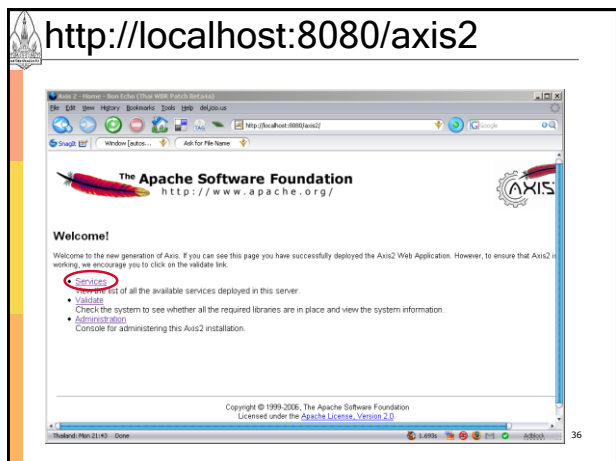
Creating axis2.war using Apache Ant

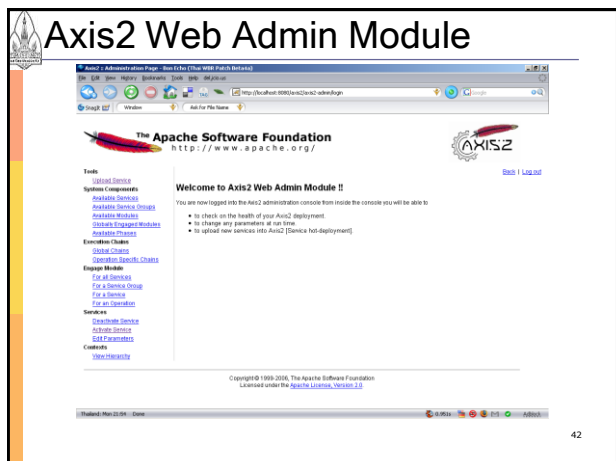
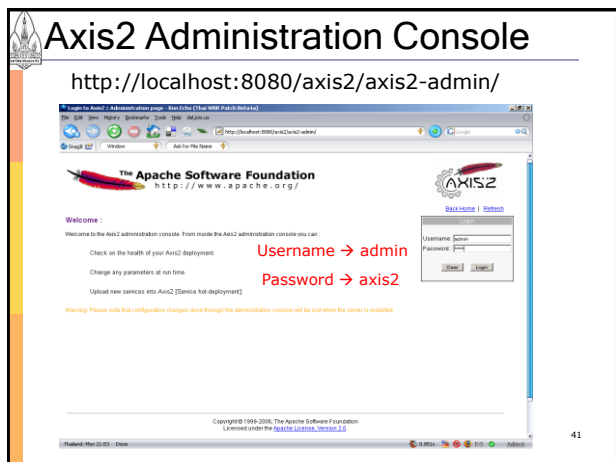
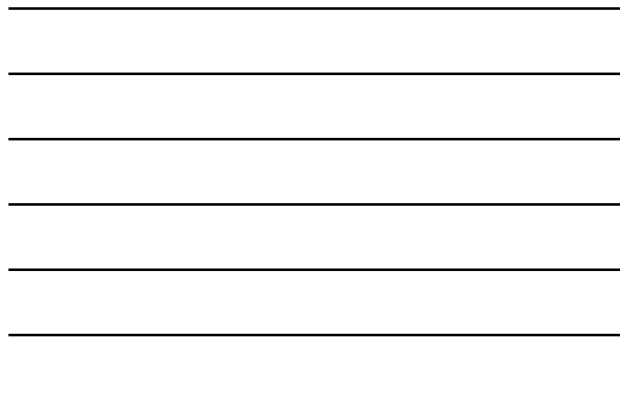
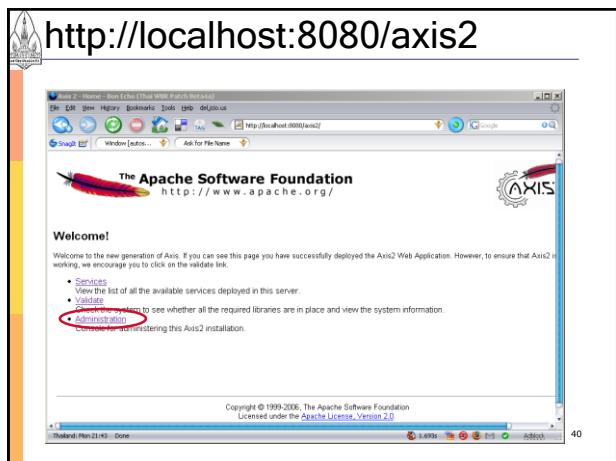
- Download Apache Ant which can be downloaded from <http://ant.apache.org/bindownload.cgi>
- Set %PATH% environment variable to %ANT_HOME%/bin
- Run ant create.war at directory %AXIS2_HOME%/webapp
 - “ant” command needs the input source file “build.xml”

33









Axis2 Services Context

- The Directory Structure of axis2.war
 - axis2-web
 - META-INF
 - WEB-INF
 - classes
 - conf
 - axis2.xml
 - lib
 - modules
 - services
 - services.list
 - ...
 - version.aar
 - web.xml

43

Axis2 Services Structure

- Services can be deployed as *.aar files
- The *.aar can be built using ant command in the directory that has build.xml

44

Deploy Sample Web Service : StockQuoteService

- Go to %AXIS2_HOME%\samples\quickstart which has the structure as follows
 - resources
 - META-INF
 - Services.xml
 - src
 - Java files
 - build.xml
- Create StockQuoteService.aar by running command "ant generate.service"

45

Create StockQuoteService.aar

```

C:\apache\axis2-1.1.1>cd %AXIS2_HOME%\samples\quickstart
C:\apache\axis2-1.1.1\samples\quickstart>ant generate.service
buildfile: build.xml

compile.service:
[mkdir] created dir: c:\apache\axis2-1.1.1\samples\quickstart\build
[mkdir] created dir: c:\apache\axis2-1.1.1\samples\quickstart\build\classes
[javac] Compiling 1 source file to c:\apache\axis2-1.1.1\samples\quickstart\build\classes
[javac] Note: C:\apache\axis2-1.1.1\samples\quickstart\src\samples\quickstart\service\pojo\stockquoteservice.java uses unchecked or unsafe operation.
[javac] Note: Recompile with -Xlint:unchecked for details.

generate.service:
[copy] Copying 1 file to c:\apache\axis2-1.1.1\samples\quickstart\build\classes
[jar] Building jar: c:\apache\axis2-1.1.1\samples\quickstart\build\stockquoteservice.aar

BUILD SUCCESSFUL
Total time: 11 seconds
  
```

Deploy StockQuoteService

Now copy StockQuoteService.aar from
 %AXIS2_HOME%\samples\quickstart\build to
 %CATALINA_HOME%\webapps\axis2\WEB-INF\services

Deployed StockQuoteService

http://localhost:8080/axis2/services/listServices

WSDL of StockQuoteService

```

<?xml version='1.0' encoding='UTF-8'>
<definitions name='StockQuoteService' targetNamespace='http://packtbot.com/stockQuoteService/'>
<import namespace='http://schemas.xmlsoap.org/soap/envelope/'/>
<binding name='StockQuoteServiceSoap' type='soap:binding' style='rpc' xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/'/>
<operation name='getPrice' xmlns:tns='http://packtbot.com/stockQuoteService/'/>
<input message='tns:getPriceRequest' style='rpc' xmlns:tns='http://packtbot.com/stockQuoteService/'/>
<output message='tns:getPriceResponse' style='rpc' xmlns:tns='http://packtbot.com/stockQuoteService/'/>
<operation name='update' xmlns:tns='http://packtbot.com/stockQuoteService/'/>
<input message='tns:updateRequest' style='rpc' xmlns:tns='http://packtbot.com/stockQuoteService/'/>
<output message='tns:updateResponse' style='rpc' xmlns:tns='http://packtbot.com/stockQuoteService/'/>
<operation name='addStock' xmlns:tns='http://packtbot.com/stockQuoteService/'/>
<input message='tns:addStockRequest' style='rpc' xmlns:tns='http://packtbot.com/stockQuoteService/'/>
<output message='tns:addStockResponse' style='rpc' xmlns:tns='http://packtbot.com/stockQuoteService/'/>
</definitions>

```

Call Stock Quote Service

□ <http://localhost:8080/axis2/rest/StockQuoteService/getPrice?symbol=IBM>

```

<ns:getPriceResponse>
  <ns:return>42.0</ns:return>
</ns:getPriceResponse>

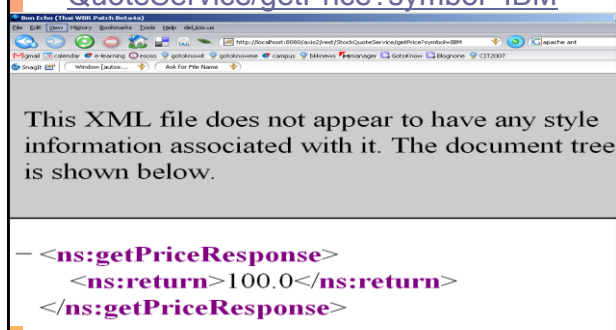
```

Update Stock Quote Price

□ <http://localhost:8080/axis2/rest/StockQuoteService/update?symbol=IBM&price=100>

Call Stock Quote Service after Update

□ <http://localhost:8080/axis2/rest/StockQuoteService/getPrice?symbol=IBM>



```

<ns:getPriceResponse>
  <ns:return>100.0</ns:return>
</ns:getPriceResponse>
    
```

Agenda

- What and Why Apache Axis2?
- Understanding AXIOM
- Learning Axis2 Basics
- Understanding the Deployment Model
- Axis2 Installation
- Generating a Service and Deploying using POJO (Plain Old Java Object)
- Generating a Service and Deploying using AXIOM

53

Create CalculatorService Overview

- Create directory CalculatorService
 - Edit Java source file
 - Edit services.xml
 - Edit build.xml
- Use ant to create CalculatorService.aar
- Copy CalculatorService.aar to %CATALINA_HOME%\webapps\axis2\WEB-INF\services

54

Create CalculatorService

- Create directory CalculatorService (May copy from StockQuoteService)
 - resources
 - META-INF
 - Services.xml
 - src
 - service
 - pojo
 - CalculatorService.java
 - build.xml
 - README.txt

Edit Java Source File (1/2)

```

package service.pojo;
public class CalculatorService {
    public int add(int a, int b) {
        return a+b;
    }
    public int sub(int a, int b) {
        return a-b;
    }
}
    
```

Edit Java Source File (2/2)

```

public int mult(int a, int b) {
    return a*b;
}

public int div(int a, int b) {
    if (b != 0)
        return a/b;
    else
        return 0;
}
}
    
```

Edit services.xml

```

<service name="CalculatorService" scope="application"
targetNamespace="http://quickstart.samples/">
  <description>
    Calculator Service
  </description>
  <messageReceivers>
    <messageReceiver mep="http://www.w3.org/2004/08/wsdl/in-only"
class="org.apache.axis2.rpc.receivers.RPCInOnlyMessageReceiver"/>
    <messageReceiver mep="http://www.w3.org/2004/08/wsdl/in-out"
class="org.apache.axis2.rpc.receivers.RPCMessageReceiver"/>
  </messageReceivers>
  <schema schemaNamespace="http://quickstart.samples/xsd"/>
  <parameter
name="ServiceClass">service.pojo.CalculatorService</parameter>
</service>
    
```

58

Edit build.xml (1/3)

```

<project basedir="." default="generate.service">
  <property environment="env"/>
  <property name="AXIS2_HOME" value="{env.AXIS2_HOME}"/>
  <property name="build.dir" value="build"/>
  <path id="axis2.classpath">
    <fileset dir="{AXIS2_HOME}/lib">
      <include name="*.jar"/>
    </fileset>
  </path>
  <target name="compile.service">
    <mkdir dir="{build.dir}"/>
    <mkdir dir="{build.dir}/classes"/>
    <!--First let's compile the classes-->
    <javac debug="on"
fork="true"
destdir="{build.dir}/classes"
srcdir="{basedir}/src"
classpathref="axis2.classpath">
    </javac>
  </target>
    
```

59

Edit build.xml (2/3)

```

<target name="generate.wsdl" depends="compile.service">
  <taskdef name="java2wsdl"
classname="org.apache.ws.java2wsdl.Java2WSDLTask"
classpathref="axis2.classpath"/>
  <java2wsdl className="service.pojo.CalculatorService"
outputLocation="{build.dir}"
targetNamespace="http://quickstart.samples/"
schemaTargetNamespace="http://quickstart.samples/xsd">
    <classpath>
      <pathelement path="{axis2.classpath}"/>
      <pathelement location="{build.dir}/classes"/>
    </classpath>
  </java2wsdl>
</target>
    
```

60

Edit build.xml (3/3)

```

<target name="generate.service" depends="compile.service">
  <!--aar them up -->
  <copy toDir="${build.dir}/classes" failonerror="false">
    <fileset dir="${basedir}/resources">
      <include name="**/*.xml"/>
    </fileset>
  </copy>
  <jar destfile="${build.dir}/CalculatorService.aar">
    <fileset excludes="**/Test.class" dir="${build.dir}/classes"/>
  </jar>
</target>

<target name="clean">
  <delete dir="${build.dir}"/>
</target>

</project>
    
```

61

Use ant to create CalculatorService.aar

- Go to directory CalculatorService
- Run command "ant generate.service"

```

C:\apache\tomcat\apache-tomcat-5.5.20\webapps\axis2\services\calculatorService>
ant generate.service
Buildfile: build.xml

compile.service:
[mkdir] Created dir: C:\apache\tomcat\apache-tomcat-5.5.20\webapps\axis2\services\calculatorService\build
[mkdir] Created dir: C:\apache\tomcat\apache-tomcat-5.5.20\webapps\axis2\services\calculatorService\build\classes
[javac] Compiling 1 source file to C:\apache\tomcat\apache-tomcat-5.5.20\webapps\axis2\services\calculatorService\build\classes

generate.service:
[copy] Copying 1 file to C:\apache\tomcat\apache-tomcat-5.5.20\webapps\axis2\services\calculatorService\build\classes
[jar] Building jar: C:\apache\tomcat\apache-tomcat-5.5.20\webapps\axis2\services\calculatorService\build\calculatorService.aar

BUILD SUCCESSFUL
Total time: 2 seconds
    
```

Now we have CalculatorService.aar in directory CalculatorService\build

62

Deploy CalculatorService

- Copy CalculatorService.aar from CalculatorService\build to %CATALINA_HOME%\webapps\axis2\WEB-INF\services

Deployed CalculatorService

The screenshot shows the Apache Axis2 web console. Under the 'Available services' section, 'CalculatorService' is listed and circled in red. Below it, the service status is 'Active' and 'Available Operations' is empty. The 'Version' section shows a 'getVersion' operation.

WSDL of CalculatorService

The screenshot shows the WSDL document tree for CalculatorService. The XML structure is visible, including the service name, port, and multiple operations like add, subtract, multiply, and divide.

Call Calculator Service (Add Operation)

□ <http://localhost:8080/axis2/rest/CalculatorService/add?a=2&b=3>

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```

- <ns:addResponse>
  <ns:return>5</ns:return>
</ns:addResponse>
    
```

Agenda

- What and Why Apache Axis2?
- Understanding AXIOM
- Learning Axis2 Basics
- Understanding the Deployment Model
- Axis2 Installation
- Generating a Service and Deploying using POJO (Plain Old Java Object)
- Generating a Service and Deploying using AXIOM

67

Generating the Service using AXIOM

- Go to directory
%AXIS2_HOME%\samples\quickstart
axiom
- Build the service
- Deploy the service

68

%AXIS2_HOME%\samples\quickstartaxiom

- quickstartaxiom
 - README.txt
 - build.xml
 - resources
 - META-INF
 - services.xml
 - StockQuoteService.wsdl
 - src
 - ..
 - service
 - ...
 - StockQuoteService.java
 - clients
 - AXIOMClient.java

69

StockQuoteService.java (1/4)

```

/* %AXIS2_HOME\samples\quickstartAxiom\src\samples\quickstart\service\axiom
\StockQuoteService.java by The Apache Software Foundation. */
package samples.quickstart.service.axiom;
import javax.xml.stream.XMLStreamException;
import org.apache.axiom.om.OMAbstractFactory;
import org.apache.axiom.om.OMElement;
import org.apache.axiom.om.OMFactory;
import org.apache.axiom.om.OMNamespace;
import java.util.HashMap;
public class StockQuoteService {
    private HashMap map = new HashMap();
    
```

70

StockQuoteService.java (2/4)

```

public OMElement getPrice(OMElement element) throws
XMLStreamException {
    element.build();
    element.detach();
    OMElement symbolElement =
element.getFirstElement();
    String symbol = symbolElement.getText();

    String returnText = "42";
    Double price = (Double) map.get(symbol);
    if(price != null){
        returnText = "" + price.doubleValue();
    }
    
```

71

StockQuoteService.java (3/4)

```

OMFactory fac = OMAbstractFactory.getOMFactory();
    OMNamespace omNs =

    fac.createOMNamespace("http://quickstart.samples/xsd"
, "ns");
    OMElement method =
fac.createOMElement("getPriceResponse", omNs);
    OMElement value = fac.createOMElement("return",
omNs);
    value.addChild(fac.createOMText(value, returnText));
    method.addChild(value);
    return method;
    }
    
```

72

StockQuoteService.java (4/4)

```

public void update(OMElement element) throws
XMLStreamException {
    element.build();
    element.detach();
    OMElement symbolElement =
element.getFirstElement();
    String symbol = symbolElement.getText();
    OMElement priceElement =
(OMElement)symbolElement.getNextOMSibling();
    String price = priceElement.getText();
    map.put(symbol, new Double(price));
}
    
```

73

services.xml

```

<service name="StockQuoteService" scope="application">
<description>
    Stock Quote Service
</description>
<operation name="getPrice">
    <messageReceiver
class="org.apache.axis2.receivers.RawXMLINOutMessageReceiver"/>
</operation>
<operation name="update">
    <messageReceiver
class="org.apache.axis2.receivers.RawXMLINOnlyMessageReceiver"/>
</operation>
<parameter
name="ServiceClass">samples.quickstart.service.axiom.StockQuoteSer
vice</parameter>
</service>
    
```

74

Build the Service using AXIOM

- Go to %AXIS2_HOME%\samples\quickstartaxiom
- Build the service using "ant" command

```

C:\apache\axis2-1.1.1\samples\quickstartaxiom>ant generate.service
Buildfile: build.xml

compile:
[mkdir] Created dir: C:\apache\axis2-1.1.1\samples\quickstartaxiom\build
[mkdir] Created dir: C:\apache\axis2-1.1.1\samples\quickstartaxiom\build\cla
sses
[javac] Compiling 2 source files to C:\apache\axis2-1.1.1\samples\quickstart
axiom\build\classes
[javac] Note: C:\apache\axis2-1.1.1\samples\quickstartaxiom\src\samples\quic
kstart\service\axiom\StockQuoteService.java uses unchecked or unsafe operations.
[javac] Note: Recompile with -Xlint:unchecked for details.

generate.service:
[copy] Copying 2 files to C:\apache\axis2-1.1.1\samples\quickstartaxiom\buil
d\classes
[jar] Building jar: C:\apache\axis2-1.1.1\samples\quickstartaxiom\build\St
ockQuoteService.aar

BUILD SUCCESSFUL
Total time: 8 seconds
    
```

Deploy the Service using AXIOM

Copy StockQuoteService.aar from
 %AXIS2_HOME%\samples\quickstartaxiom\dist
 to
 %CATALINA_HOME%\webapps\axis2\WEB-INF\services

The screenshot shows two Windows Explorer windows. The left window shows the source directory: C:\Program Files\Apache Software Foundation\Axis2\dist. The right window shows the destination directory: C:\Program Files\Apache Software Foundation\Axis2\WEB-INF\services. The file 'StockQuoteService.aar' is being copied from the source to the destination.

Check Stock Quote Service Schema

http://localhost:8080/axis2/services/StockQuoteService?xsd

The screenshot shows a web browser window with the URL http://localhost:8080/axis2/services/StockQuoteService?xsd. The page content displays the XML Schema Definition (XSD) for the service. The schema includes elements for 'getStockQuote' and 'getStockQuoteResponse'.

AXIOMClient.java (1/5)

```

    /* %AXIS2_HOME%\samples\quickstart\Axiom\src\samples\quickstart\clients\
    AXIOMClient.java by The Apache Software Foundation. */
    package samples.quickstart.clients;

    import org.apache.axiom.om.OMAbstractFactory;
    import org.apache.axiom.om.OMElement;
    import org.apache.axiom.om.OMFactory;
    import org.apache.axiom.om.OMNamespace;
    import org.apache.axis2.Constants;
    import org.apache.axis2.addressing.EndpointReference;
    import org.apache.axis2.client.Options;
    import org.apache.axis2.client.ServiceClient;
    
```

AXIOMClient.java (2/5)

```

public class AXIOMClient {
    private static EndpointReference targetEPR =
        new EndpointReference(
            "http://localhost:8080/axis2/services/StockQuoteService");
    public static OMElement getPricePayload(String symbol) {
        OMFactory fac = OMAbstractFactory.getOMFactory();
        OMNamespace omNs = fac.createOMNamespace(
            "http://quickstart.samples.xsd", "tns");

        OMElement method = fac.createOMEElement("getPrice", omNs);
        OMElement value = fac.createOMEElement("symbol", omNs);
        value.addChild(fac.createOMText(value, symbol));
        method.addChild(value);
        return method;
    }
}
    
```

79

AXIOMClient.java (3/5)

```

public static OMElement updatePayload(String symbol, double price) {
    OMFactory fac = OMAbstractFactory.getOMFactory();
    OMNamespace omNs = fac.createOMNamespace(
        "http://quickstart.samples.xsd", "tns");
    OMElement method = fac.createOMEElement("update", omNs);

    OMElement value1 = fac.createOMEElement("symbol", omNs);
    value1.addChild(fac.createOMText(value1, symbol));
    method.addChild(value1);

    OMElement value2 = fac.createOMEElement("price", omNs);
    value2.addChild(fac.createOMText(value2,
        Double.toString(price)));
    method.addChild(value2);
    return method;
}
    
```

80

AXIOMClient.java (4/5)

```

public static void main(String[] args) {
    try {
        OMElement getPricePayload =
            getPricePayload("WSO");
        OMElement updatePayload =
            updatePayload("WSO", 123.42);
        Options options = new Options();
        options.setTo(targetEPR);

        options.setTransportInProtocol(Constants.TRANSPORT
            _HTTP);

        ServiceClient sender = new ServiceClient();
        sender.setOptions(options);
    }
}
    
```

81

AXIOMClient.java (5/5)

```

sender.fireAndForget(updatePayload);
    System.err.println("done");
    Thread.sleep(3000);
    OMElement result =
sender.sendReceive(getPricePayload);

    String response = result.getFirstElement().getText();
    System.err.println("Current price of WSO: " +
response);

} catch (Exception e) {
    e.printStackTrace();
}
}
    
```

Generating the Client using AXIOM

- Go to directory
%AXIS2_HOME%\samples\quickstartaxiom
- Build and run the client

```

Directory of C:\apache\axis2-1.1.1\samples\quickstartaxiom
01/15/2007 04:04 PM <DIR> .
01/15/2007 04:04 PM <DIR> ..
01/15/2007 04:04 PM <DIR> build
01/09/2007 06:42 PM 2,353 build.xml
01/09/2007 06:42 PM 1,050 README.txt
01/15/2007 11:42 AM resources
01/15/2007 11:42 AM <DIR> src
2 File(s) 3,403 bytes
5 Dir(s) 5,451,481,088 bytes free

C:\apache\axis2-1.1.1\samples\quickstartaxiom>ant run.client
Buildfile: build.xml

compile:
run.client:
[java] done
[java] Current price of WSO: 123.42

BUILD SUCCESSFUL
Total time: 7 seconds
C:\apache\axis2-1.1.1\samples\quickstartaxiom>
    
```

CalculatorServiceAxiom


[CalculatorServiceAxiom](#)
 Service EPR : <http://localhost:8080/axis2/services/CalculatorServiceAxiom>
 Service REST epr : <http://localhost:8080/axis2/rest/CalculatorServiceAxiom>

Service Description : Calculator Service Axiom

Service Status : Active

Available Operations

- mult
- add
- div
- sub

 **References**

- The Apache Software Foundation, "Apache Axis2/Java", <http://ws.apache.org/axis2/>, Retrieved on January 9, 2007
- Deepal Jayasinghe and Ruchith Fernando, "Building Enterprise Applications with Axis2"
- Chathura Herath and Eran Chinthaka, "Axis 2 Tutorial"

88
