Judd Solutions

Development, Mentoring and Training

Consuming and Producing Web Services with Web Tools

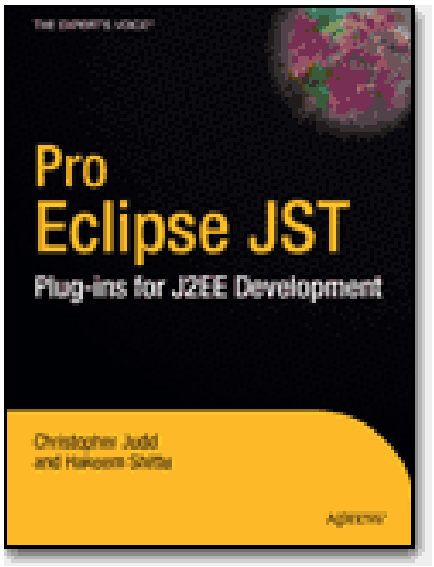
Christopher M. Judd

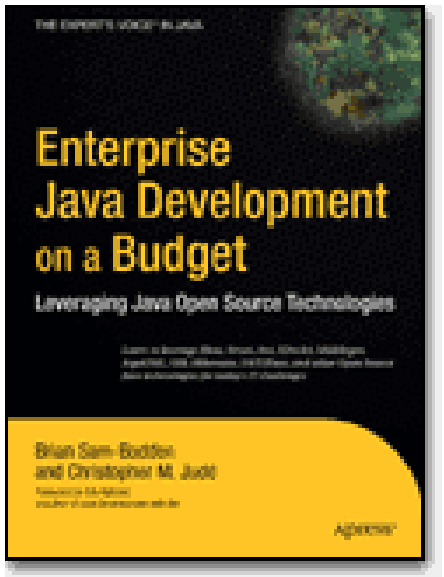
President/Consultant

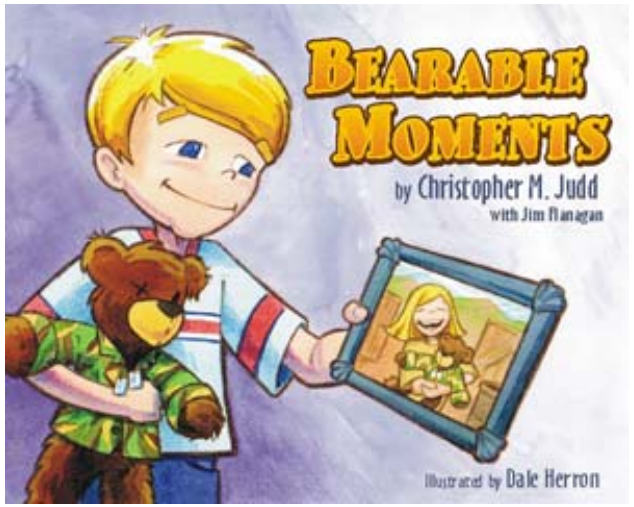
Judd Solutions, LLC

Christopher M. Judd

- President/Consultant of Judd Solutions
- Central Ohio Java User Group (COJUG) coordinator







Other Sessions of Interest

- WTP
  - T-4. Develop Better J2EE Applications with the Web Tools Platform (Wed 8:45)
  - 101. Quick Tour of the Eclipse Web Tools Platform (Thu 8:30)
  - 107. Leveraging JSF Components (Thu 8:30)
  - 201. How to Build Java Web Applications with the Web Tools Platform (Thu 10:30)
  - 301. Facing JavaServer Faces Development with JSF Tools (Thu 1:15)
  - 401. Consuming and Producing Web Services with Web Tools (Thu 3:15)
  - 501. Developing Java Web Services with the Web Tools Platform (Fri 8:45)
  - 504. Developing Rich Applications with JSF and AJAX (Fri 8:45)
  - 701. Building Applications with the Java Persistence API and Dali (Fri 1:45)
  - 706. Step by Step: Making Enterprise JavaBeans with J2EE Standard Tools (Fri 1:45)
  - 801. How to Use and Extend Eclipse's XML and Schema Tools (Fri 3:45)
- Other
  - T-6. Implementing SOA in Eclipse (Wed 8:45)
  - T-7. Callisto Boot Camp: Ten Projects. One Day (Wed 8:45)
  - 106. How to Improve Database Connectivity with the Data Tools Platform (Thu 8:30)
  - 107. Leveraging JSF Components (Thu 8:30)
  - 202. Web 2.0 the Eclipse Way with the Rich AJAX Platform (Thu 10:30)
  - 204. Interacting with Relational Databases (Thu 10:30)
  - 406. Developing and Deploying Services using the SOA Tools Platform (Thu 406)
  - 607. Advanced User Interface Programming Using the Eclipse Rich Client Platform (Fri 10:45)

Agenda

- WTP Overview
- Web Services Overview
- Consuming
- Producing Web Services (bottom up)
- Testing
- Deployment
- Conclusion

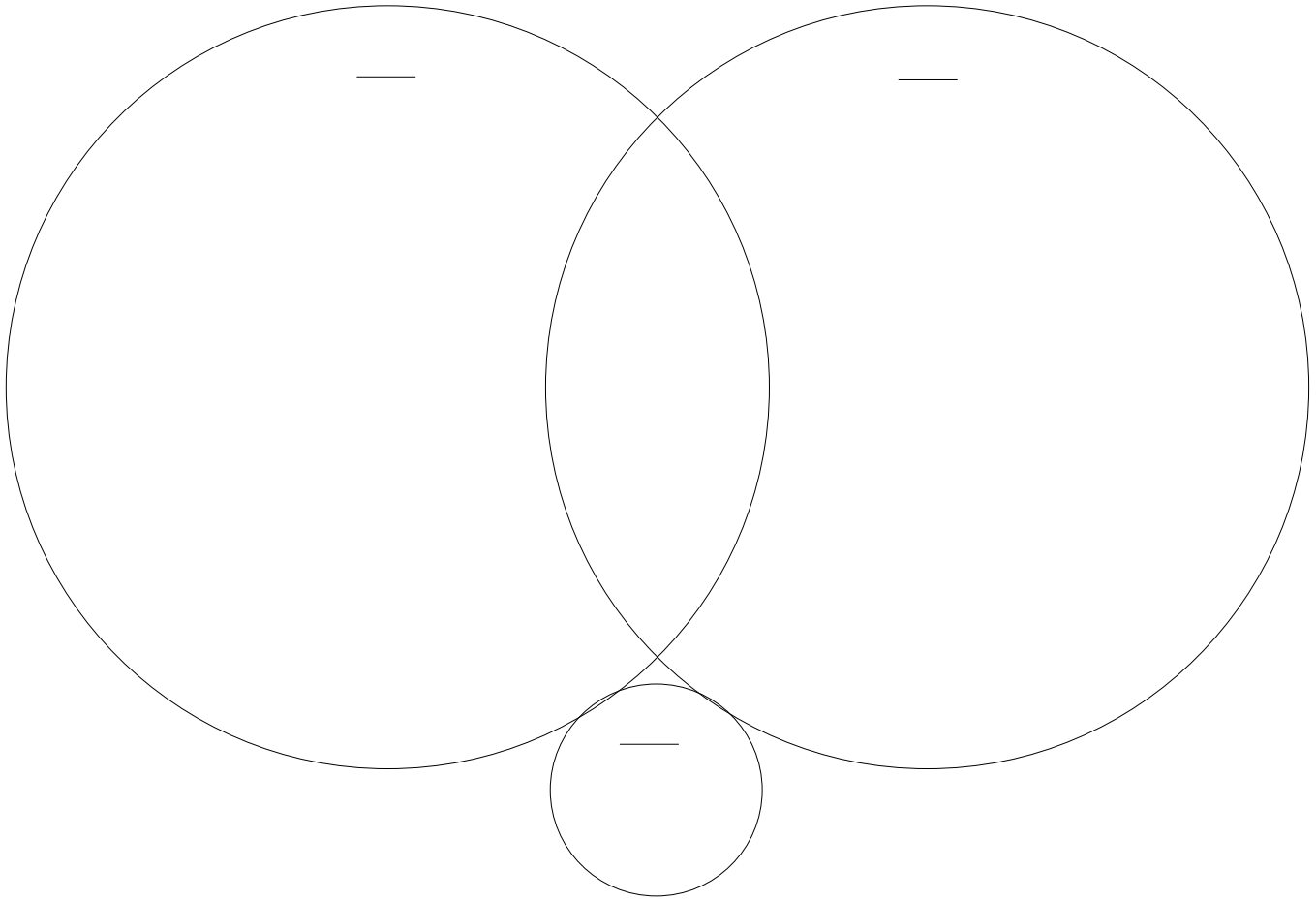
Web Tools Platform Project (WTP)

- Top Level Eclipse Project
- www.eclipse.com/webtools/
- Included in Callisto release
- Development tools for web and JEE development
  - No runtime dependencies
  - Vendor extensible
- Subprojects
  - Web Standard Tools (WST)
  - J2EE Standard Tools (JST)
  - JavaServer Faces Tools (JSF)
- Current version 1.5
- Dependencies
  - Eclipse 3. 2
  - Eclipse Modeling Framework (EMF) 2.2.0
  - Graphic Editor Framework (GEF) 3.2
  - Java EMF Model (JEM) 1.2

WTP Subprojects

- Web Standard Tools (WST)
  - Web artifacts
  - Defined by open standards bodies
- J2EE Standard Tools (JST)
  - J2EE components
  - Java Community Process (JCP)
  - Depends on WST
- JavaServer Faces Tools (JSF)

WTP Scope



WTP Web Services Scope

- World Wide Web Consortium (W3C)
  - http://www.w3.org
  - HTML, XHTML, CSS, XML, XSLT, XML Schema, XML Query
- Organizations for Advancement of Structured Information Standards (OASIS)
  - http://www.oasis-open.org
  - e-Business standards for web services
- Web Services Interoperability Organizations (WS-I)
  - http://www.ws-i.org
  - Interoperable message exchange between web services
- Java Community Process (JCP)
  - http://www.jcp.org
  - Java Web Services APIs

WTP Installation Options

- Callisto
- All-in-one
  - Eclipse 3.2
  - Eclipse Modeling Framework (EMF) 2.2.0
  - Graphic Editor Framework (GEF) 3.2
  - Java EMF Model (JEM) 1.2
- Update Manager
  - Eclipse.org update site
- Piecemeal

* JSF is a separate install

Agenda

- WTP Overview
- Web Services Overview
- Consuming
- Producing Web Services (bottom up)
- Testing
- Deployment
- Conclusion

Simple Object Access Protocol

From the draft W3C specification:

> SOAP is a lightweight protocol for exchange of information in a decentralized, distributed environment. It is an XML based protocol that consists of three parts: 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 datatypes, and a convention for representing remote procedure calls and responses.

http://www.w3.org/TR/soap/

WSDL

- Web Service Definition Language
- Describes
  - What the service can do
  - Where it resides
  - How to invoke it
- Elements
  - Types – data type definition
  - Message – definition of data being communicated
  - Port Type – abstract set of operations
  - Binding – concrete protocol and data format
  - Service – collection of related endpoints
  - Port – binding and a network address

Apache Axis

- Apache Web Services Project
- Open Source
- SOAP implementation
- Version 1.3.0
- http://ws.apache.org/axis/

Agenda

- WTP Overview
- Web Services Overview
- Consuming
- Producing Web Services (bottom up)
- Testing
- Deployment
- Conclusion

Weather Forecast Web Service

- Forecasting weather by zip code
  - 7 day forecast
  - Place name
  - State code
  - Latitude and Longitude
- Found at X Methods (www.xmethods.net)
- Service description
  - www.webservicex.net/WeatherForecast.asmx

Zip code: 43085 Submit Query

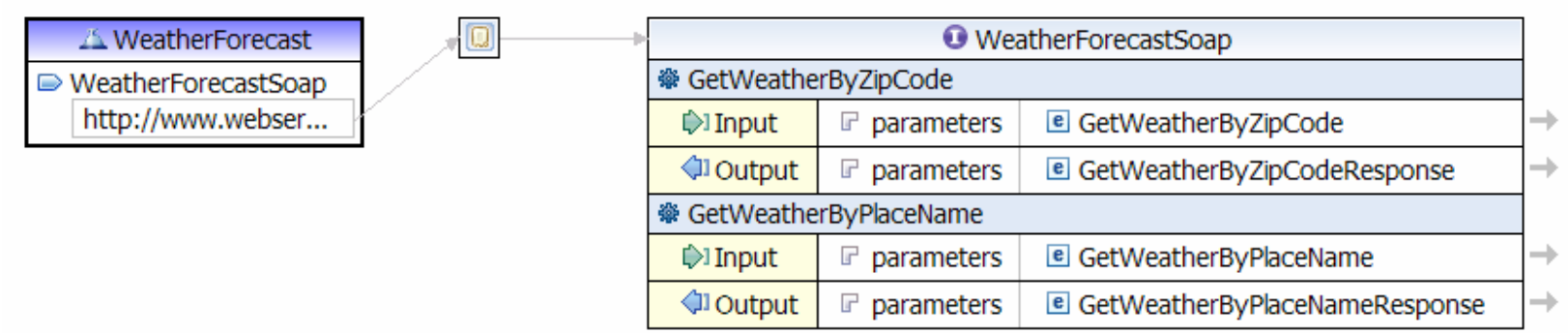
Place Names: WORTHINGTON
State Code: OH
Area Code: 614
Time Zone: EST
Latitude: 40.105156
Longitude: 83.01007
Allocation Factor: 0.00251
FIPS Code: 39

7 Day Weather Forcast:

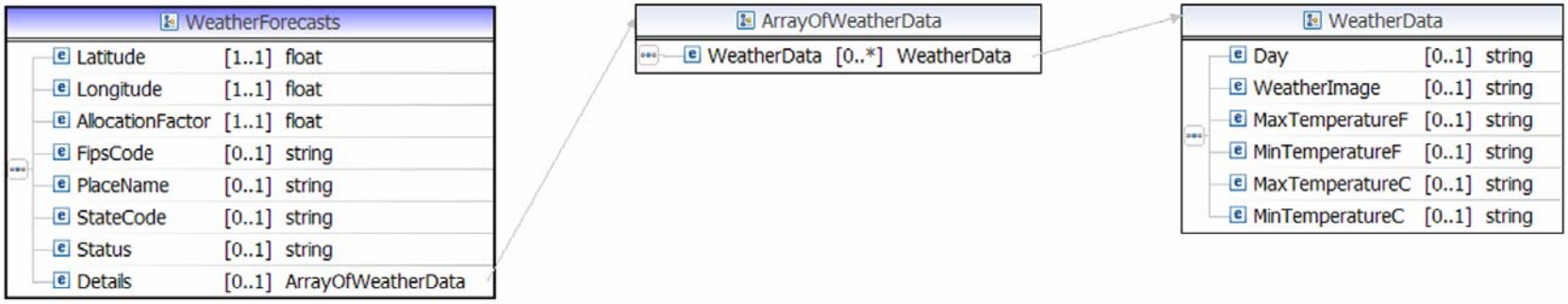
Saturday, July 22, 2006		80°F/57°F 27°C/14°C
Sunday, July 23, 2006		86°F/61°F 30°C/16°C
Monday, July 24, 2006		87°F/65°F 31°C/18°C
Tuesday, July 25, 2006		87°F/68°F 31°C/20°C
Wednesday, July 26, 2006		85°F/68°F 29°C/20°C
Thursday, July 27, 2006		85°F/68°F 29°C/20°C
Friday, July 28, 2006		85°F/68°F 29°C/20°C

Weather Forecast Web Service

Service description



Data transfer objects



Consuming Web Services Strategy

A common anti-pattern is to generate the web service client stub code directly into an application. Instead generate the code in its own project, export it as a jar and manage it as a jar.

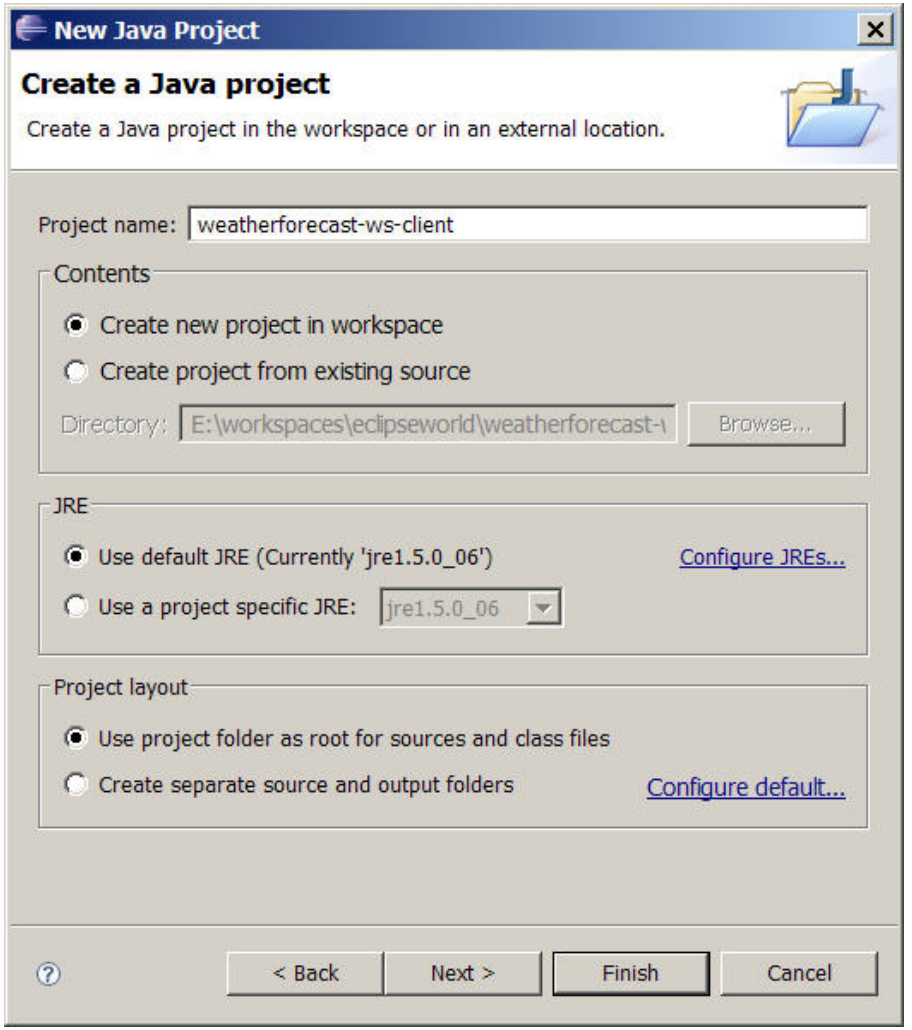
- Less files to maintain
- Improves reuse of client code
- Generated code does not follow code quality standards

Consuming Web Services Steps

1. Create Java Project
2. Download WSDL
3. Generate Web Service Client from WSDL
4. Export client Jar

Create Project

- Create a standard Java Project
- File > New > Project > Java Project

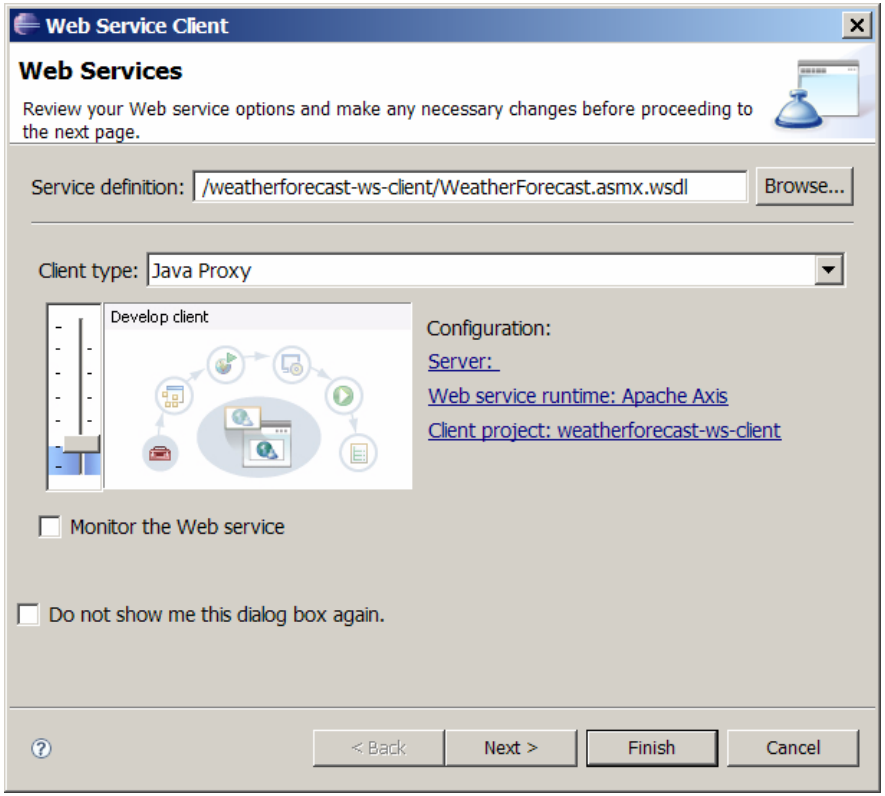


Download WSDL

- Download WSDL
- Add WSDL to project
- Include it in the exported Jar file
- Provides traceability incase there is a need to know which WSDL version was used to generate stubs and Jar

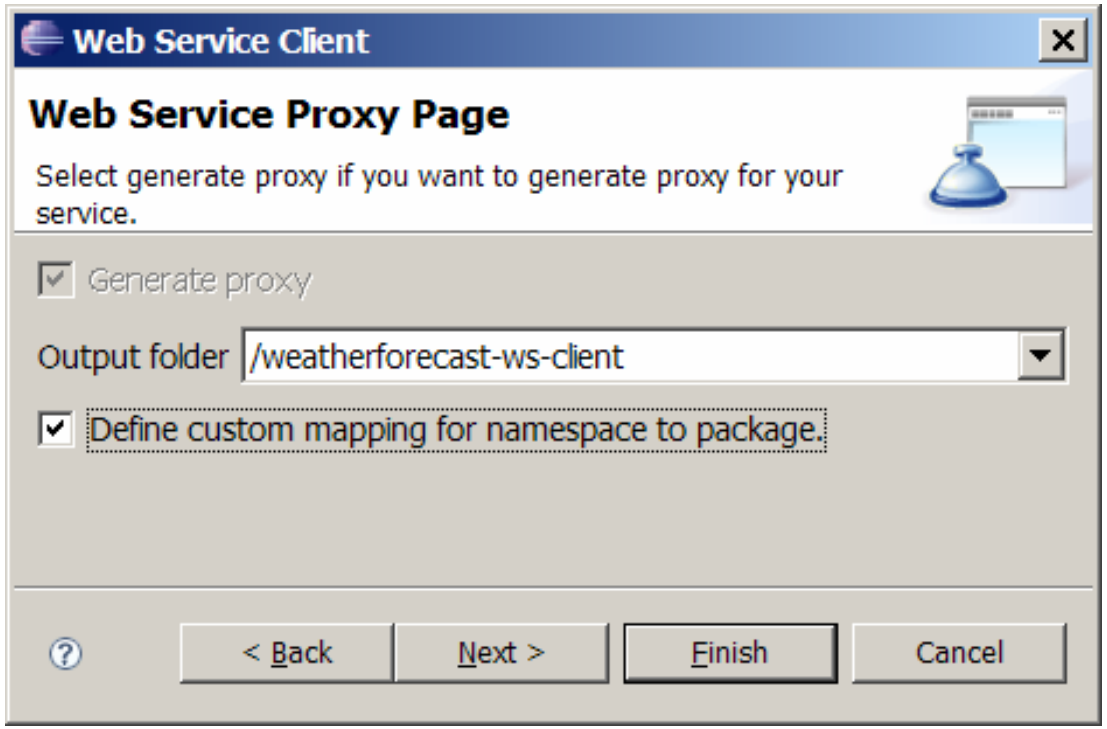
Generate Web Service Client

- Right click on WSDL and choose Web Services > Generate Client
  or
  File > New > Web Services > Web Service Client
- Specify Java Proxy
- Monitor traffic using TCP/IP Monintor



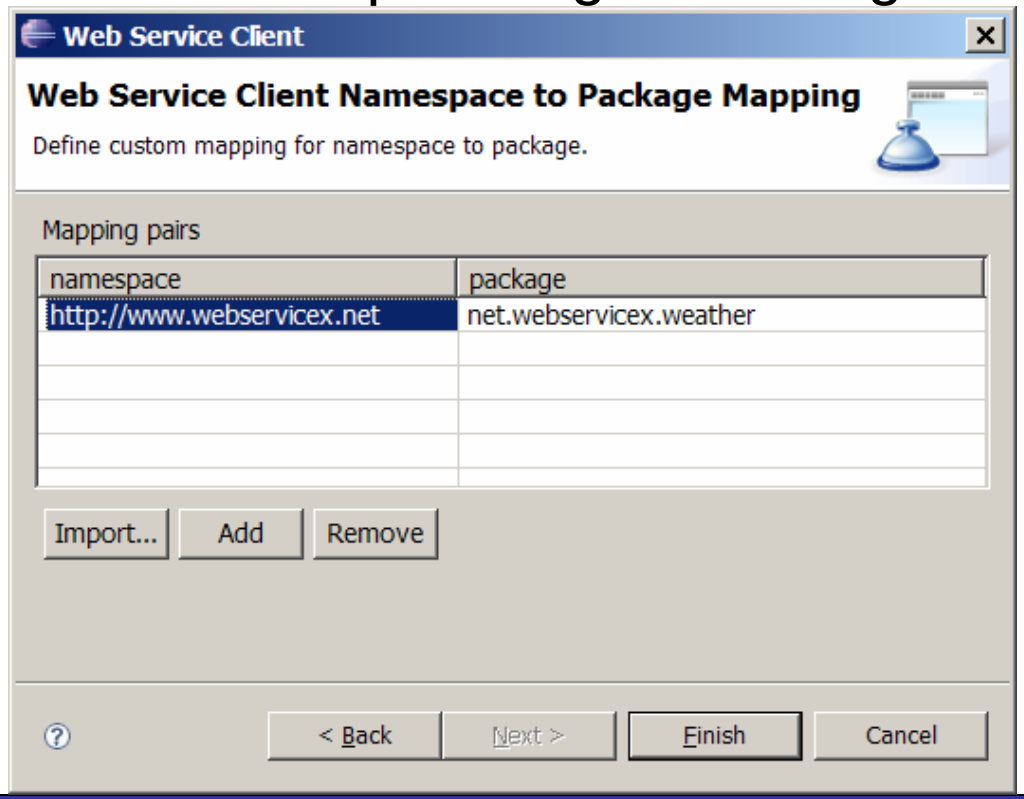
Generate Web Service Client

- Specify project to generate the code in



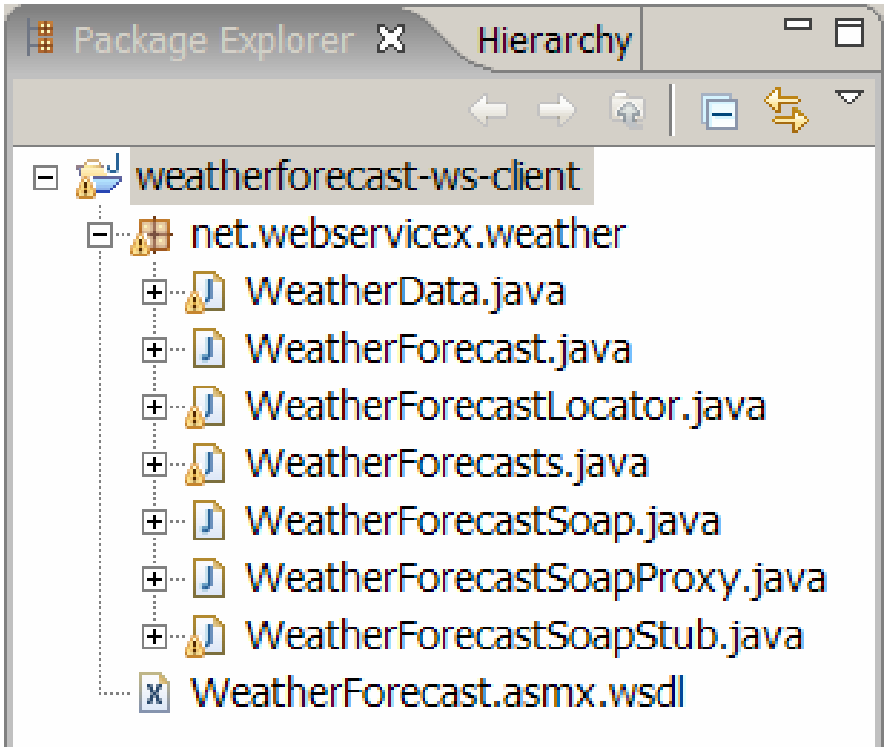
Generate Web Service Client

- Optionally, can specify namespace to package mapping
- Convert targetNamespace to a more conventional Java package naming convention.



Output

- Classes
  - Service interface
  - Dynamic proxy service impl
  - Stub service impl
  - Service locator interface
  - Service locator impl
  - Data transfer objects
- Axis Jars
  - axis.jar
  - jaxrpc.jar
  - saaj.jar
  - wsdl4j-1.5.1.jar
  - commons-discovery-0.2.jar
  - commons-logging-1.0.4.jar



* When using the client Jar, these Axis jars will be required as well.

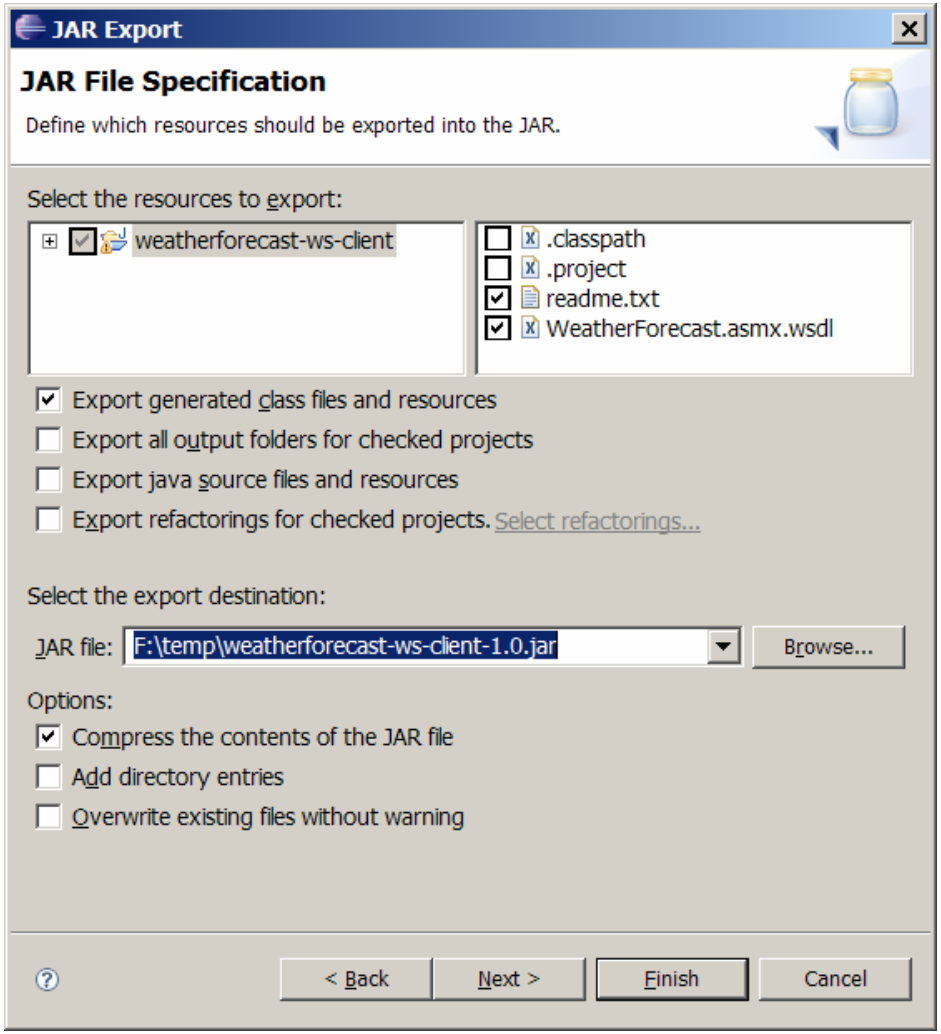
Using Generate Classes


```
WeatherForecastLocator forcastLocator = new WeatherForecastLocator();
WeatherForecastSoap wfSoap = forcastLocator.getWeatherForecastSoap();
WeatherForecasts forecasts = wfSoap.getWeatherByZipCode(zipCode);
WeatherData[] details = forecasts.getDetails();

System.out.println(forecasts.getLatitude());
System.out.println(forecasts.getLongitude());

for (int i = 0; i < details.length; i++) {
    WeatherData detail = details[i];
    System.out.println(detail.getDay());
    System.out.println(detail.getWeatherImage());
    System.out.println(detail.getMaxTemperatureF());
}
```

Export client Jar



Agenda

- WTP Overview
- Web Services Overview
- Consuming
- Producing Web Services (bottom up)
- Testing
- Deployment
- Conclusion

Producing Web Services

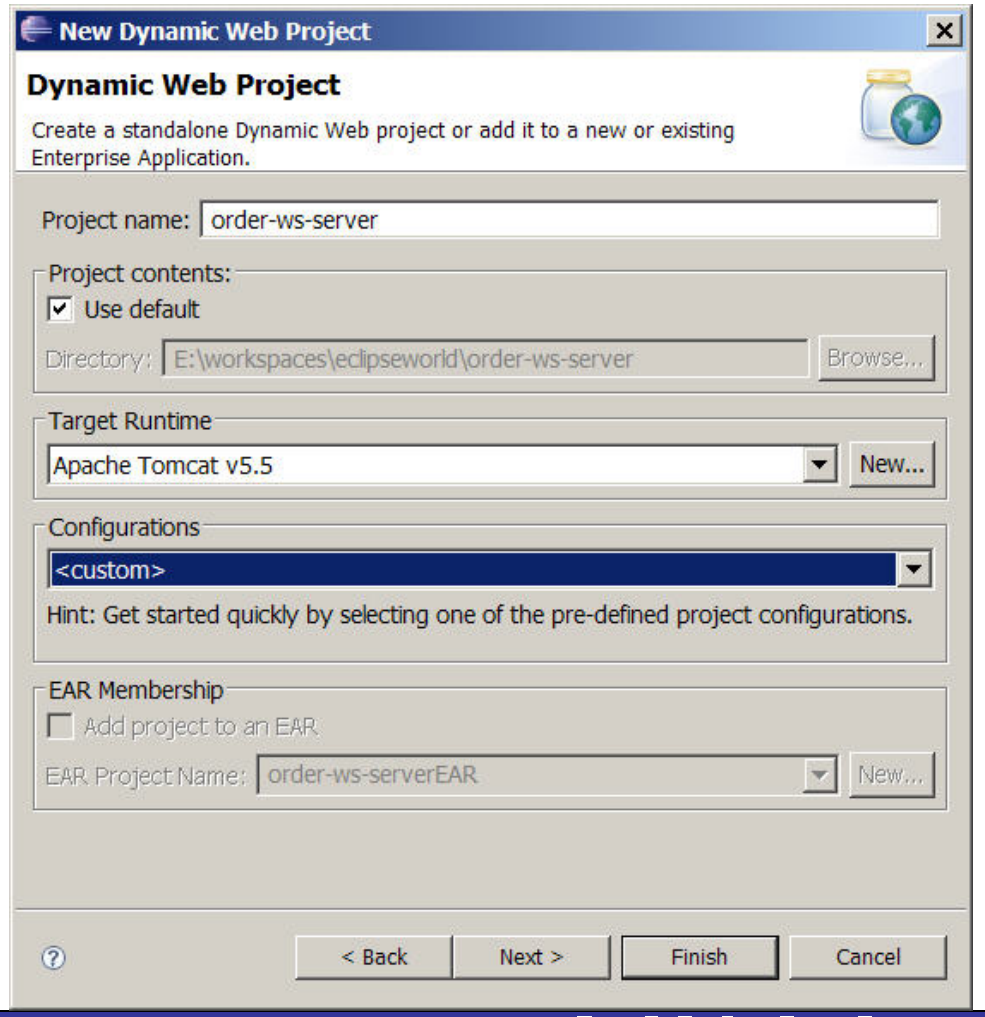
- Bottom up
  - Java code -> Generate WSDL
- Top down
  - WSDL -> Generate Java code

Bottom Up Steps

1. Create Dynamic Web Project
2. Create Service and DTOs
3. Generate Web Service

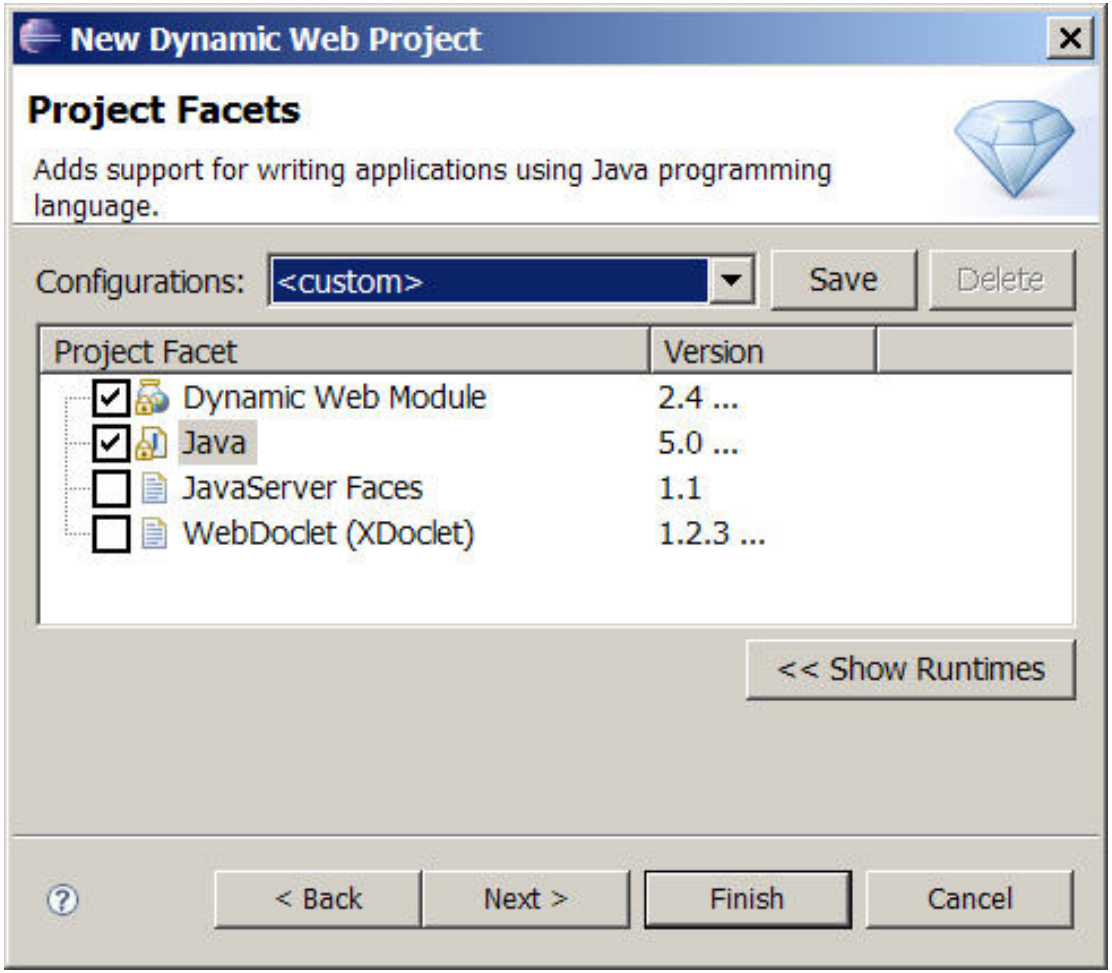
Create Dynamic Web Project

- File > New > Other > Web > Dynamic Web Project
- Project Name
- Target server



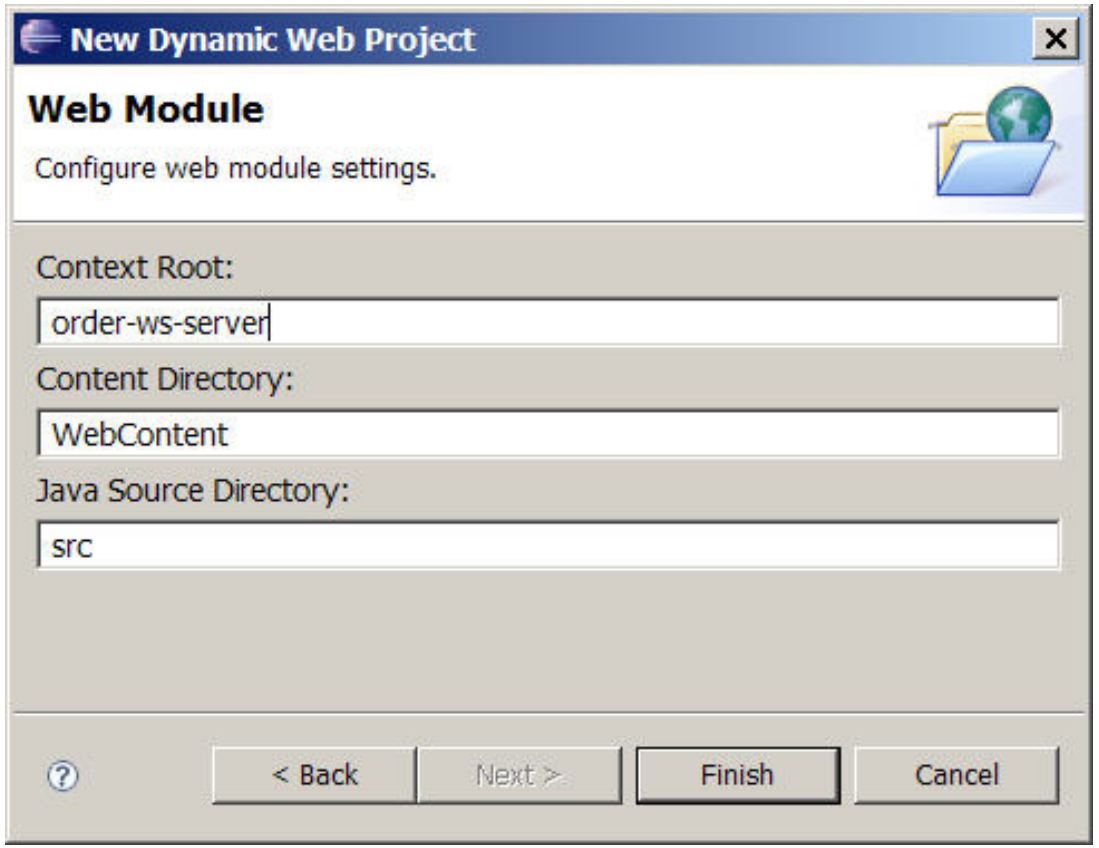
Create Dynamic Web Project

- Facets and versions



Create Dynamic Web Project

- Context Root
- Directories



Create Service

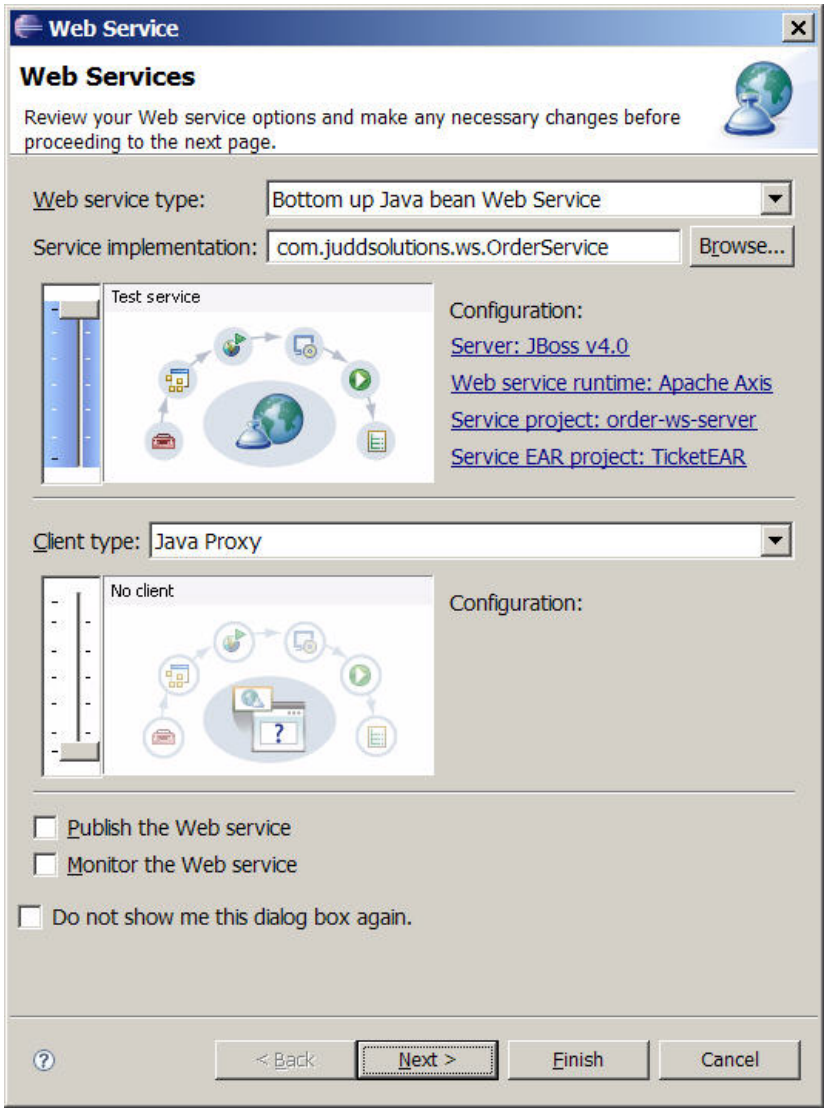
```
public class OrderService {

    private static final float DISCOUNT = 0.9f;

    public float quote(float price, int quantity) {
        if (quantity > 100) {
            price = price * DISCOUNT;
        }
        return price * quantity;
    }
}
```

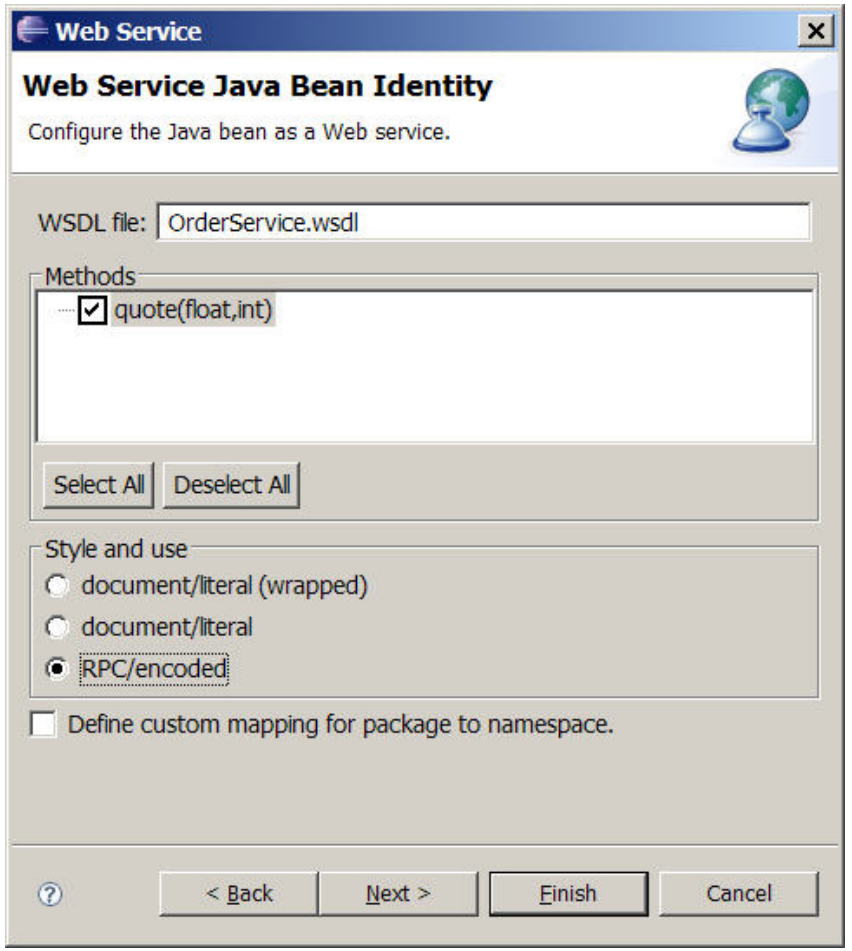
Generate Web Service

- New > File > Other > Web Services > Web Service
  or
  Right click on Class: Web Services > Create Web Services
- Select the amount to be generated
- Monitor using TCP/IP Monitor



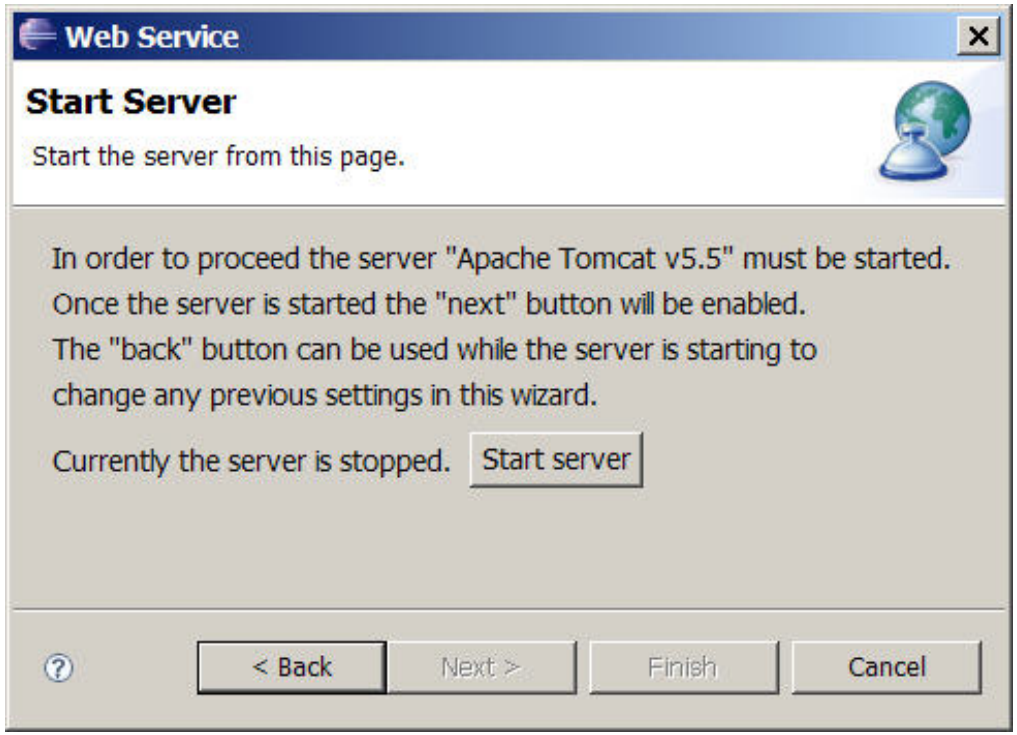
Generate Web Service

- Select methods to expose
- Select type
  - RPC
    - Performance
    - Simple types
  - Document
    - Interoperability
    - XML



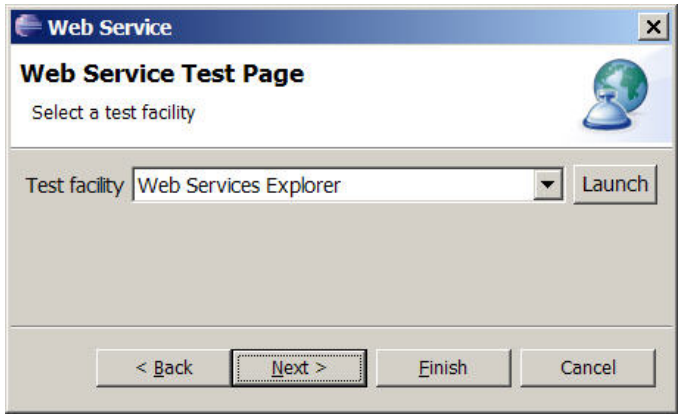
Generate Web Service

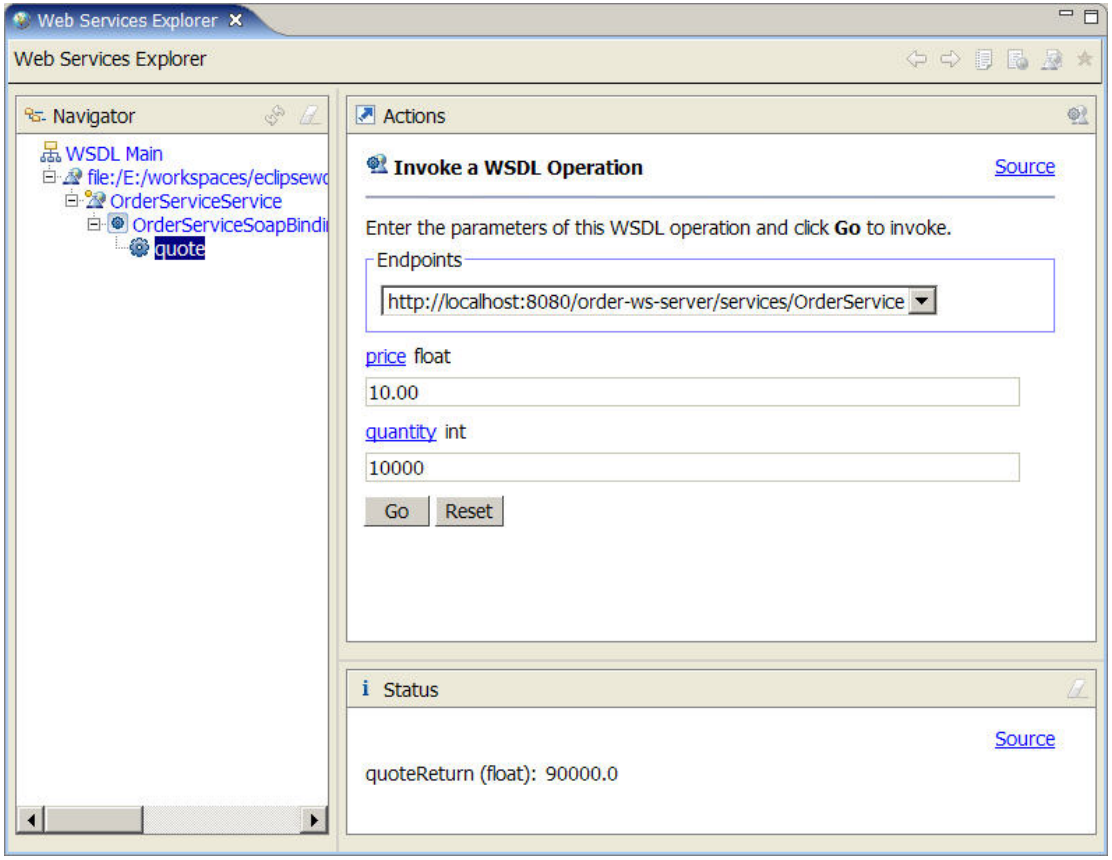
- Start server to local application service



Generate Web Service

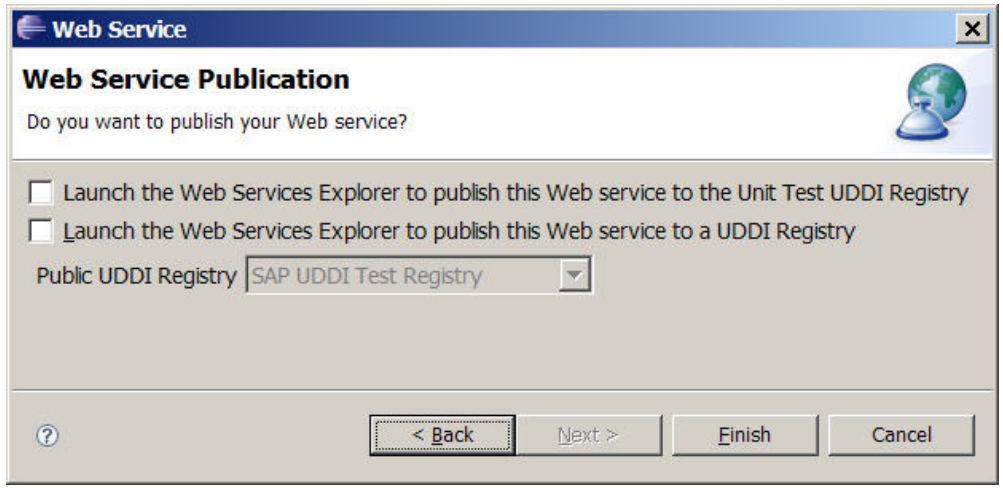
- Test Client





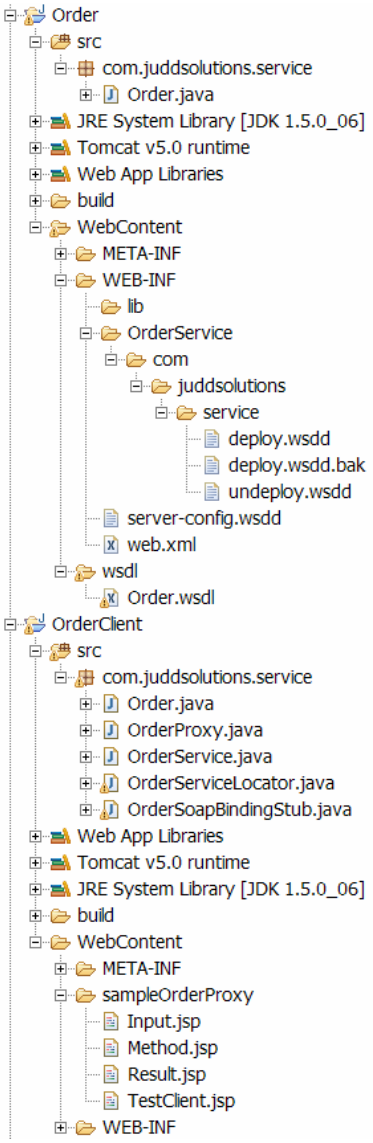
Generate Web Service

- Publish web service to UDDI registry



Output

- Web Service
  - Axis deployment descriptors
  - WSDL
- Test Client
  - Dynamic web app
  - Proxy
  - JSPs



Data Types

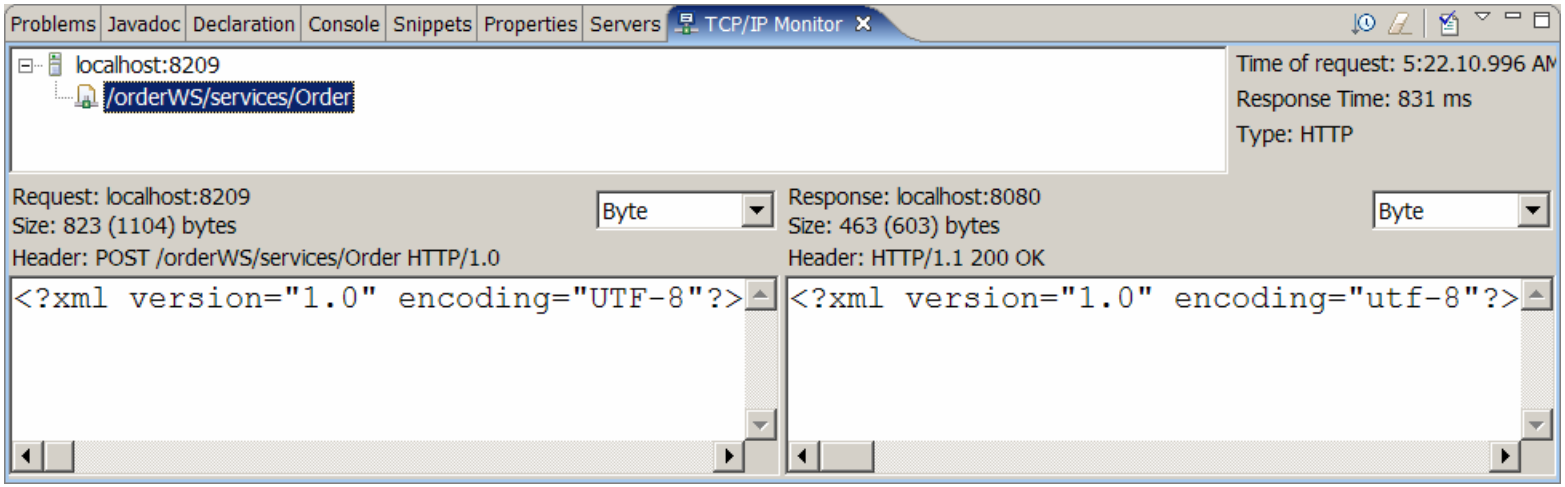
- Simple Java data types
- JavaBeans
  - No parameter constructor
- Arrays

Agenda

- WTP Overview
- Web Services Overview
- Consuming
- Producing Web Services (bottom up)
- Testing
- Deployment
- Conclusion

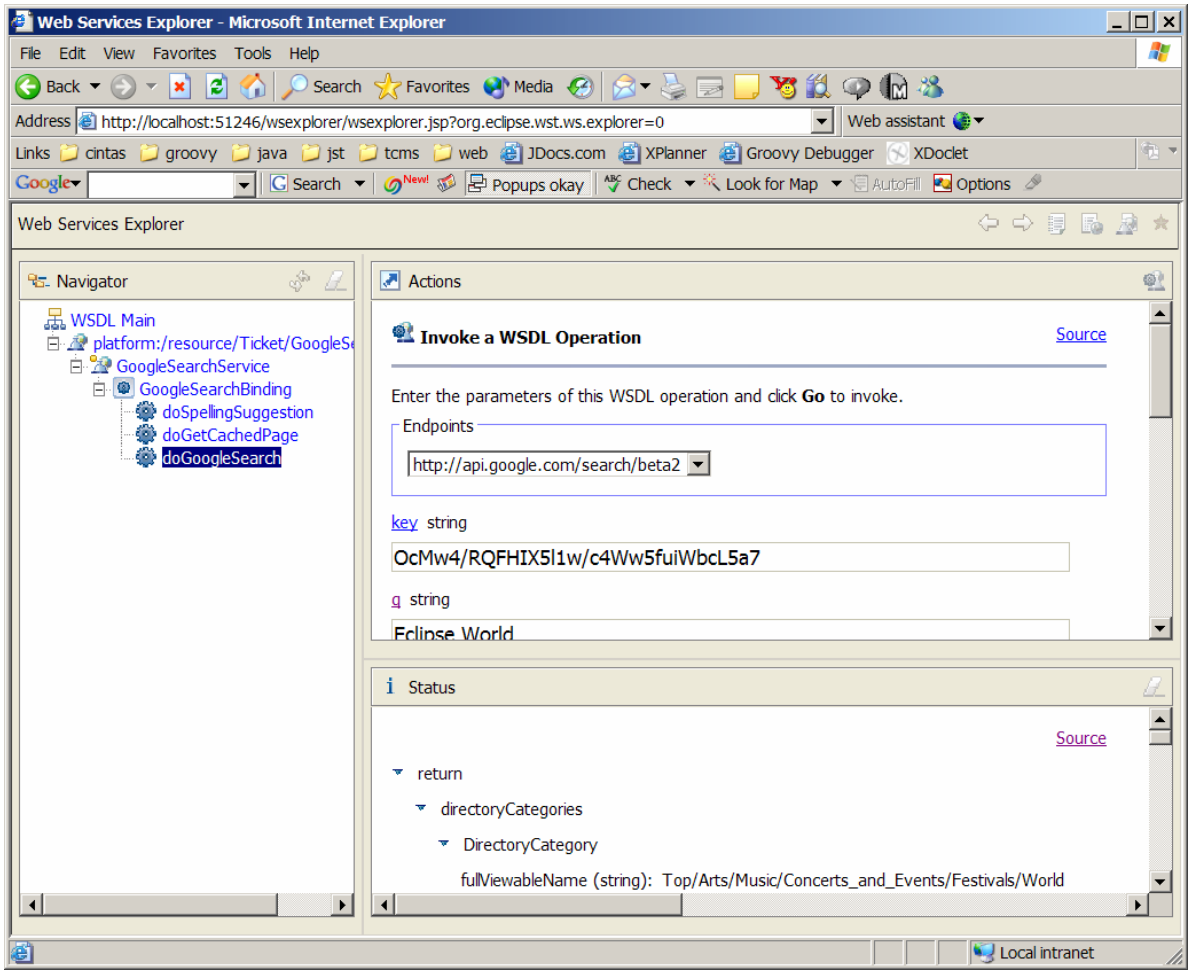
TCP/IP Monitoring

- View SOAP request and response



Web Service Explorer Testing

- Test any Web Services



Browser Testing

- Invoke Axis service via URL

Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://localhost:8080/order-ws-server/services/OrderService?method=quote&price=10&quantity=10

Go

cintas groovy Groovy Debugger maven java jst tcms web Wiki XPlanner Selling to AAFES Home ... Thistle Class Association

http://localh...0&quantity=10

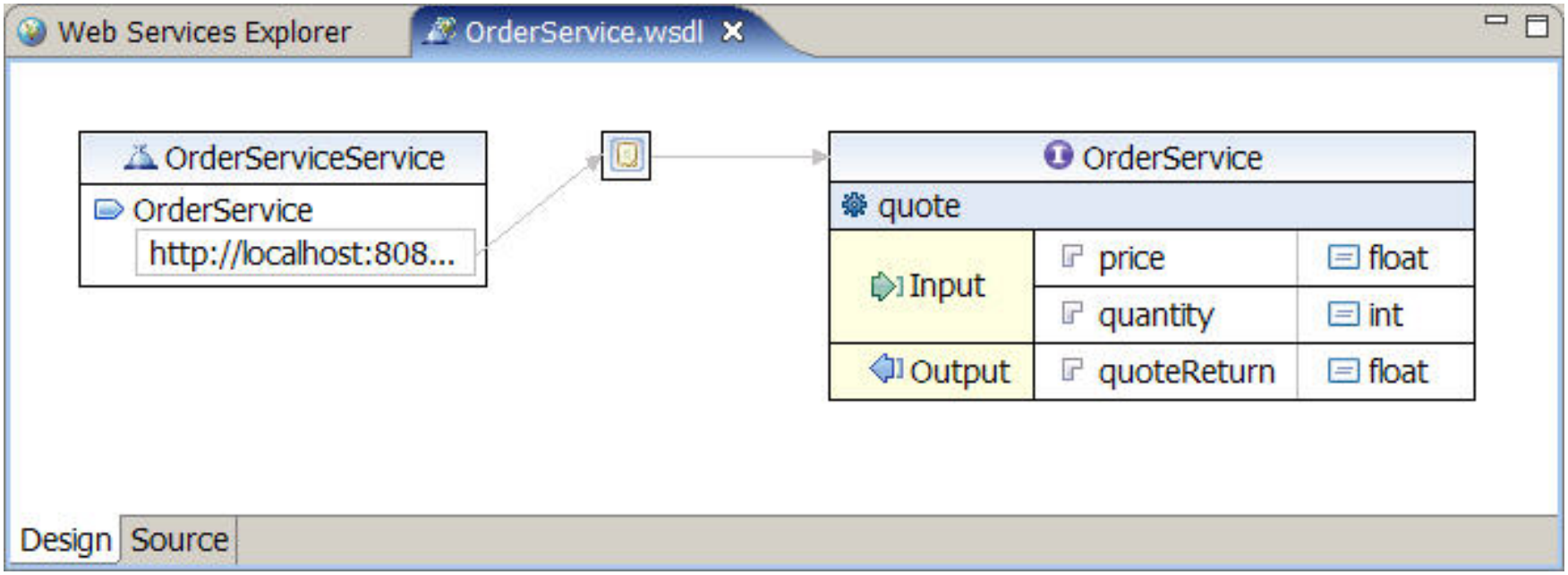
This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
- <soapenv:Envelope>
  - <soapenv:Body>
    - <quoteResponse soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
        <quoteReturn href="#id0"/>
      </quoteResponse>
      <multiRef id="id0" soapenc:root="0" soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xsi:type="xsd:float">100.0</multiRef>
    </soapenv:Body>
  </soapenv:Envelope>
```

Done

WSDL Editor

- Graphically edit WSDL

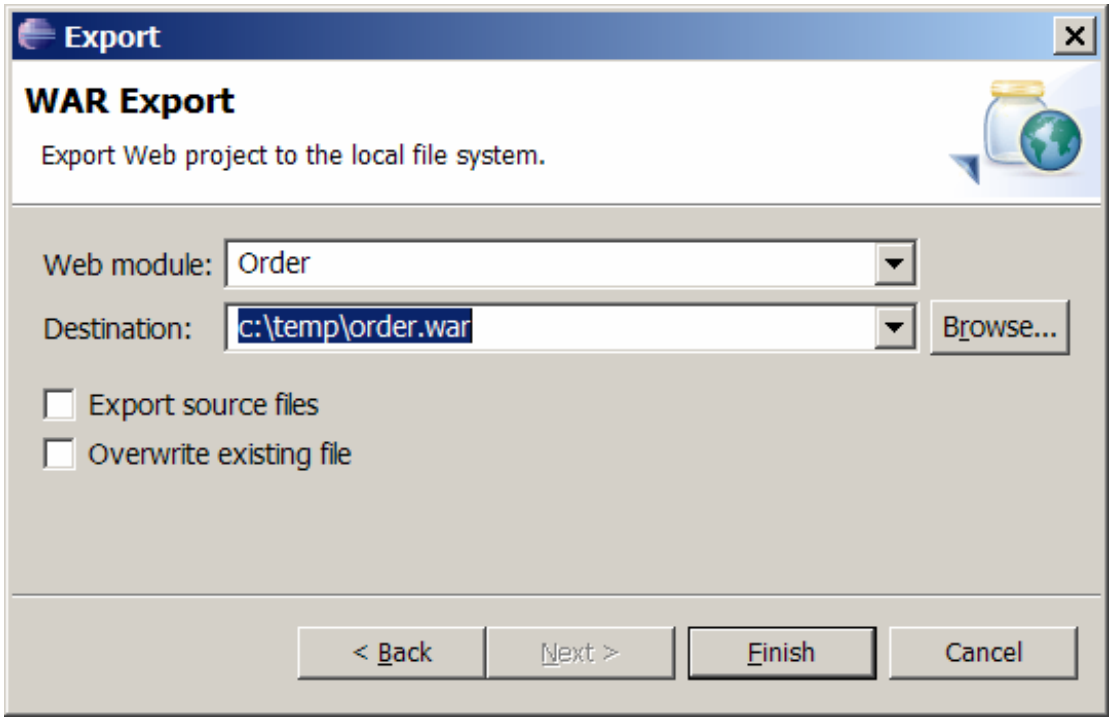


Agenda

- WTP Overview
- Web Services Overview
- Consuming
- Producing Web Services (bottom up)
- Testing
- Deployment
- Conclusion

Package

- File > Export > WAR file



Deployment

- Container specific
- Admin console
- Deployment directory
- Ant task
- Server View

Agenda

- WTP Overview
- Web Services Overview
- Consuming
- Producing Web Services (bottom up)
- Testing
- Deployment
- Conclusion

WTP Resources

- www.eclipse.org/webtools/
- www.projst.com
- Tutorials
  - http://www.eclipse.org/webtools/community/communityresources.html#tutorials
- Articles
  - http://www.eclipse.org/webtools/community/communityresources.html#articles
  - Build rich Internet applications - http://www-128.ibm.com/developerworks/edu/os-dw-os-laszlo-i.html
- New Group
  - news://news.eclipse.org/eclipse.webtools

Contact Information

- http://www.juddsolutions.com
- cjudd@juddsolutions.com
- Blog
  - http://blogs.apress.com/authors.php?author=Christopher%20Judd
- Pro Eclipse JST
  - http://www.projst.com
  - http://www.apress.com/book/bookDisplay.html?bID=447
- Enterprise Java Development on a Budget
  - http://www.apress.com/book/bookDisplay.html?bID=197

Questions ?

Please fill out your evaluations.

Slides can be found at

http://www.juddsolutions.com/ew2006/