Consuming and Producing Web Services with WST and JST

Christopher M. Judd
President/Consultant
Judd Solutions, LLC
Christopher M. Judd

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

• Long Talks
  - Developing Data Tools with the DTP SQL Models and Parsers (Tues. 10:45)
  - EJB 3.0 Persistence and the Dali EJB ORM Project (Tues. 2:15)
  - What’s New in Web Tools 1.0 and 1.5 (Thurs. 9:45)
  - Using and Extending the Eclipse Web Tools Platform (WTP) (Thurs. 10:45)
  - Build JavaServer™ Faces Applications with the JSF Tools Project (Thurs. 2:15)

• Short Talks
  - Managing APIs with the Eclipse API Scanner (Wed. 4:33)
  - The AJAX Toolkit Framework (Wed. 4:51)
  - Authoring in Eclipse (Thurs. 1:27)
  - Eclipse and Apache Derby (Thurs. 1:36)

What’s new in WTP? Find out at the WTP project sprint!
Mon. 6:30-8:30pm, room 203
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/
• Development tools for web development
  - No runtime dependencies
  - Vendor extensible
• Subprojects
  - Web Standard Tools (WST)
  - J2EE Standard Tools (JST)
  - JavaServer Faces Tools (JSF)
• Version 1.0 was released Dec 20, 2005.
• Dependencies
  - Eclipse 3.1.2
  - Eclipse Modeling Framework (EMF) 2.1.2
  - Graphic Editor Framework (GEF) 3.1.1
  - Java EMF Model (JEM) 1.1.0.1
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

• All-in-one
  - Eclipse 3.1.2
  - Eclipse Modeling Framework (EMF) 2.1.2
  - Graphic Editor Framework (GEF) 3.1.1
  - Java EMF Model (JEM) 1.1.0.1

• Update Manager
  - Eclipse.org update site

• Piecemeal
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 1.2.1
- http://ws.apache.org/axis/
Agenda

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

• Web Service for searching

  - Download developer’s kit
    • WSDL file
    • Java wrapper library
    • API documentation
  - Create a Google account
    • License key
Consuming steps

1. Create Java Project
2. Generate Web Service Client from WSDL
Create Project

- Create a standard Java Project
- File > New > Project > Java Project
Generate Web Service Client

- File > New > Other > Web Services > Web Service Client
- Specify Java Proxy
- Test application
- Monitor traffic
Generate Web Service Client

• Enter location of WSDL locally or via URL
Generate Web Service Client

- Specify project to generate the code in
Generate Web Service Client

- Optionally, can specify namespace to package mapping
Output

- **Classes**
  - DictionaryCategory
  - GoogleSearchBindingStub
  - GoogleSearchPort
  - GoogleSearchPortProxy
  - GoogleSearchResult
  - GoogleSearchService
  - GoogleSearchServiceLocator
  - ResultElement

- **Axis Jars**
  - axis.jar
  - jaxrpc.jar
  - saaj.jar
  - wsdl4j-1.5.1.jar
  - commons-discovery-0.2.jar
  - commons-logging-1.0.4.jar
Using Generate Classes

GoogleSearchPortProxy proxy = new GoogleSearchPortProxy();
GoogleSearchResult result = proxy.doGoogleSearch(
    "OcMw4/RQFHIX511w/c4Ww5fu*****", // key
    "Bearable Moments", // query terms
    0, // start index
    10, // max results (<= 10)
    true, // filter similar results
    ", // subset restrict
    true, // safe search
    ", // language restrict
    "UTF-8", // input encoding
    "UTF-8"); // output encoding

ResultElement[] resultElements = result.getResultElements();
for (int i = 0; i < resultElements.length; i++) {
    ResultElement element = resultElements[i];
    System.out.println(element.getTitle());
    System.out.println("  " + element.getURL());
}
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

![Web Module window](image)
public class Order {

    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
• Right click on Class > Web Services > Create Web Services

[Image of a software window demonstrating the process of generating a web service]
Generate Web Service

- Select class containing services to expose if not selected already
Generate Web Service

- Select project for deployment descriptors
- Create new client project
Generate Web Service

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

- Start server to host web service and test client

In order to proceed the server "Apache Tomcat v5.0" must be started.
Once the server is started the "next" button will be enabled.
The "back" button can be used while the server is starting to change any previous settings in the wizard.
Currently the server is stopped. Start server
Generate Web Service

- Test early
Generate Web Service

- Directory for client code
- Optionally, provide namespace to package mapping
Generate Web Service

- Methods to test from test client
Generate Web Service

- Publish web service to UDDI registry
Generate Web Service

- Test Client
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
Test Client

TCP/IP Monitoring

- View SOAP request and response
Web Service Explorer Testing

- Test any Web Services
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
• New Group
  – news://news.eclipse.org/eclipse.webtools
Help improve the Eclipse WTP user experience!

- **What is this study about?**
  - UBC and IBM are conducting a usage study to improve the organization of commands and functions in Eclipse based projects.

- **How can you take part?**
  - Simply install the study plug-in and use Eclipse WTP for your daily work.

For more information, or to sign up, please visit: http://www.cs.ubc.ca/~lkf/antoine
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
• Enterprise Java Development on a Budget

© 2004 Judd Solutions, LLC. All rights reserved.
Questions?

Please fill out your evaluations.