

# Judd Solutions

Development, Mentoring and Training



## T4 – Develop Better J2EE Applications with the Web Tools Platform

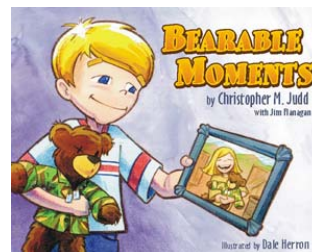
Christopher M. Judd  
President/Consultant  
Judd Solutions, LLC



Christopher M. Judd



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



Judd Solutions



## 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-7. Callisto Boot Camp: Ten Projects. One Day (Wed 8:45)
  - 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)

Judd Solutions



## Agenda

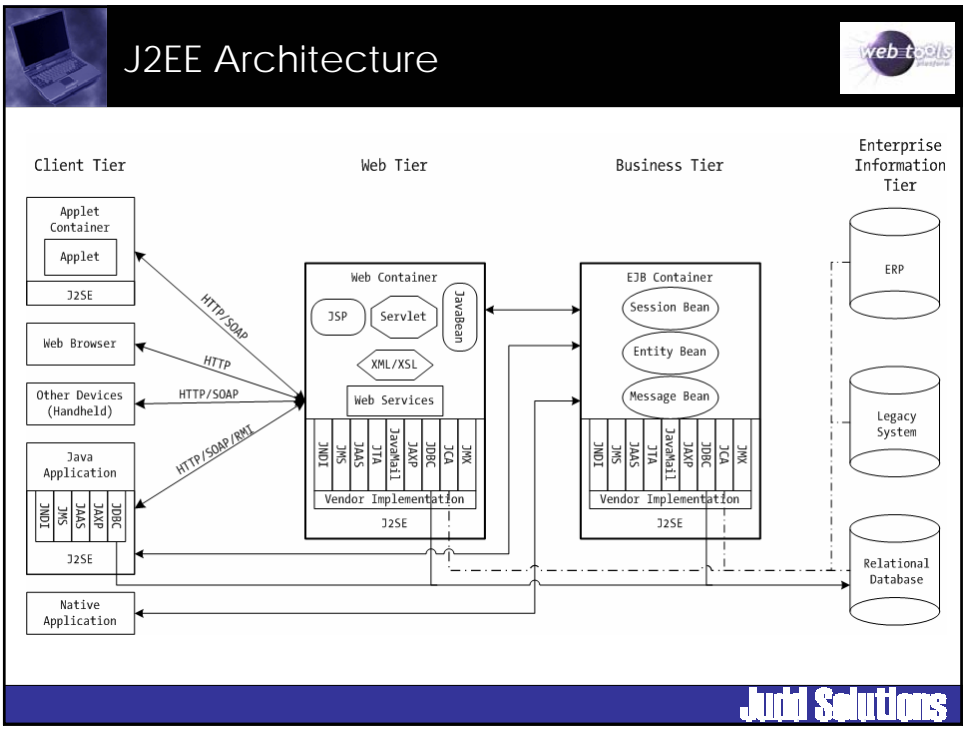


- J2EE Landscape
- WTP Overview
- WST Overview
- JST Overview
- EJBs
- Web Artifacts
- Web Services
- Packaging
- Conclusion

Judd Solutions



# J2EE Development is Complicated!!!



## Current J2EE Tool Landscape



- IBM WebSphere Application Developer (WSAD)/Rational Application Developer (RAD)
- Borland JBuilder
- Genuitec MyEclipse
- NetBeans
- JetBrains IntelliJ IDEA

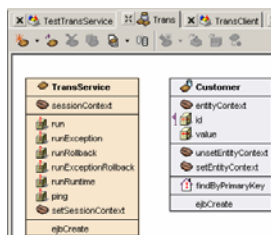
Judd Solutions



## Managing Complexity



- Code classes and interfaces by hand
- Diagrams



- Annotations

```

/**
 * A generated session bean
 * @ejb.bean name="TransService"
 *           description="A session bean named TransService"
 *           display-name="TransService"
 *           jndi-name="TransService"
 *           type="Stateless"
 *           transaction-type="Container"
 */
public abstract class TransServiceBean implements
    javax.ejb.SessionBean {

```

Judd Solutions



## Agenda



- J2EE Landscape
- WTP Overview
- WST Overview
- JST Overview
- EJBs
- Web Artifacts
- Web Services
- Packaging
- Conclusion

Judd Solutions



## Web Tools Platform Project (WTP)



- Top Level Eclipse Project
- [www.eclipse.com/webtools/](http://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

Judd Solutions



## WTP Subprojects

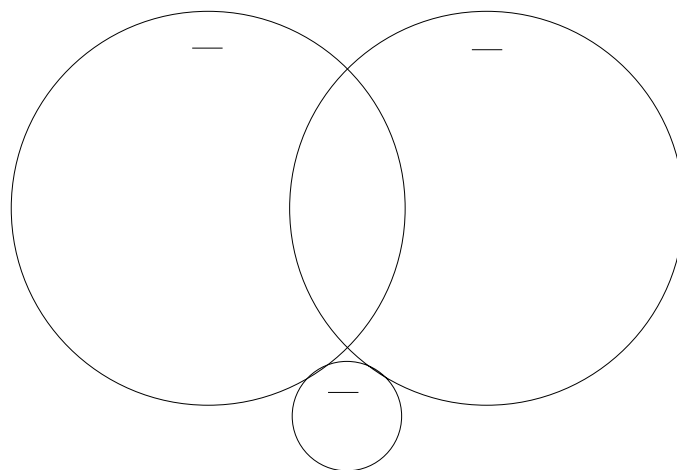


- Web Standard Tools (WST)
  - Web artifacts
  - Defined by open standards bodies
- J2EE Standard Tools (JST)
  - J2EE components
  - Java Community Process (JCP)
  - Depends on WST
- JavaServer Faces Tools (JSF)

Judd Solutions



## WTP Scope



Judd Solutions



## 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

**Judd Solutions**



## WTP Vendor Support



- IBM
- BEA
- Oracle
- Borland
- JBoss
- ObjectWeb
- Exadel
- Genuitec
- Versant

**Judd Solutions**



## Agenda



- J2EE Landscape
- WTP Overview
- WST Overview
- JST Overview
- EJBs
- Web Artifacts
- Web Services
- Packaging
- Conclusion

Judd Solutions




## WST Scope




- World Wide Web Consortium (W3C)
  - <http://www.w3.org>
  - HTML, XHTML, CSS, XML, XSLT, XML Schema, XML Query
- European Computer Manufacturers Association (ECMA)
  - <http://www.ecma-international.org>
  - JavaScript and C#
- International Engineering Task Force (IETF)
  - <http://www.ietf.org>
  - TCP/IP
- American National Standards Institute (ANSI)
  - <http://www.ansi.org>
  - SQL, C/C++
- 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

Judd Solutions

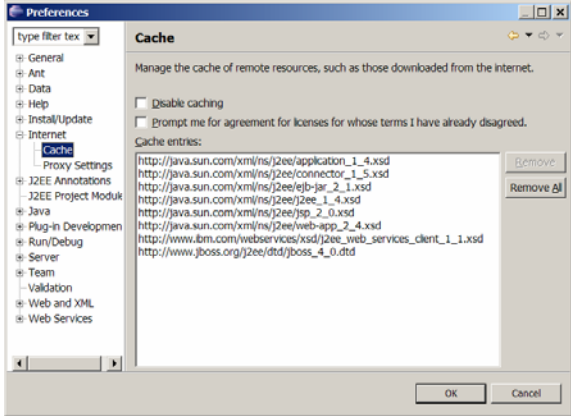





## Caching




- XML Schema & DTDs caching
- Window > Preferences > Internet > Cache



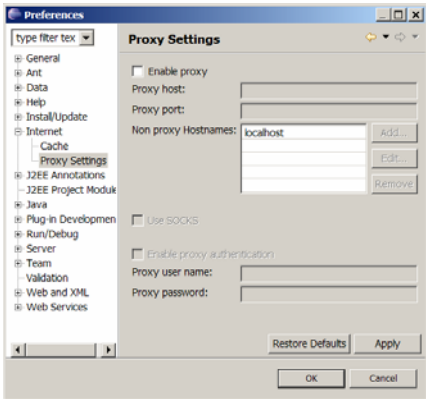
Judd Solutions



## Proxy Settings



- Enable plug-ins to use a proxy to get through the firewall
- Window > Preferences > Internet > Proxy Settings



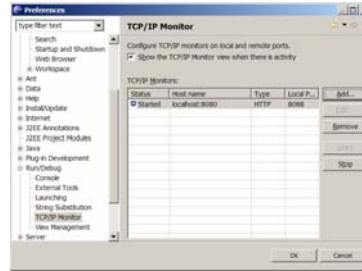
Judd Solutions



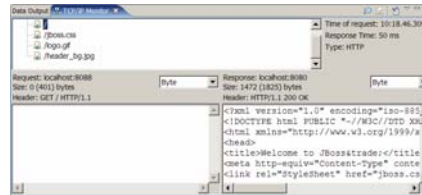
## TCP/IP Monitor



- Window > Preferences > Run/Debug > TCP/IP Monitor



- Window > Show View > Other > Debug > TCP/IP Monitor



Judd Solutions



## Structured Source Editors



- Web and XML artifacts
  - HTML, JSP, CSS, XML, XSD, DTD
- Features
  - Syntax Highlighting
  - Content Assist
  - Error Highlighting
  - Delimiter Matching
  - Source Formatting
  - Inline editing of languages
  - Templates

Judd Solutions



## XML Editor



xml	version="1.0" encoding="UTF-8"
application	(((description*, display-name*, icon*)), module+, security-role*)
id	Application_ID
version	1.4
xmlns	http://java.sun.com/xml/ns/j2ee
xmlns:xsi	http://www.w3.org/2001/XMLSchema-instance
xsi:schemaLocation	http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/application_1_4.x
display-name	TicketEAR
module	((connector   ejb   java   web), alt-dd?)
ejb	Ticket.jar

Judd Solutions



## XML Schema Editor



- Graphically or via source edit XML Schema files

Schema : http://java.sun.com/xml/ns/j2ee

Directives

- j2ee\_1\_4.xsd

Elements

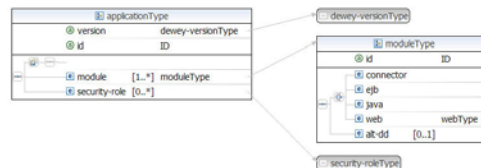
- application : applicationType

Types

- applicationType
- moduleType
- webType

Attributes

Groups



Judd Solutions

**Web Service Tools**

- WSDL Editor
- Java to WSDL and WSDL to Java wizards

The screenshot displays the WSDL Editor interface. On the left, a WSDL file named 'WeatherForecast' is loaded from 'http://www.webs...'. The main area shows the generated SOAP operations for 'WeatherForecastSoap':

Operation	Input	Output
GetWeatherByZipCode	parameters	GetWeatherByZipCodeResponse
GetWeatherByPlaceName	parameters	GetWeatherByPlaceNameResponse

Below this, the data types are defined:

- WeatherForecasts**: A list of weather forecast entries with fields like Latitude, Longitude, AllocationFactor, FipsCode, PlaceName, StateCode, Status, and Details (Array of WeatherData).
- ArrayOfWeatherData**: An array of WeatherData objects.
- WeatherData**: A single weather data entry with fields like Day, WeatherImage, MaxTemperatureF, MinTemperatureF, MaxTemperatureC, and MinTemperatureC.

**Judd Solutions**

**Test Web Service**

- Web Service Explorer can be used to test out Web Services

The screenshot shows the Web Services Explorer tool within Microsoft Internet Explorer. The 'Navigator' pane on the left shows a tree view of available services, including 'GoogleSearchService'. The 'Actions' pane shows the 'Invoke a WSDL Operation' dialog for the 'GoogleSearchService'. The 'Endpoints' field is set to 'http://ap.google.com/search/beta2'. The 'Input' field contains the string 'http://ap.google.com/search/beta2'. The 'Status' pane shows the response from the service:

```

return
  directoryCategories
    DirectoryCategory
      fullViewableName (string): Top/Arts/Music/Concerts_and_Events/Festivals/World
  
```

**Judd Solutions**



## Agenda



- J2EE Landscape
- WTP Overview
- WST Overview
- JST Overview
- EJBs
- Web Artifacts
- Web Services
- Packaging
- Conclusion

Judd Solutions



## JST Scope



WTP 1.5 focuses on J2EE 1.4 support

- EJB 2.1
- Servlet 2.4
- JSP 2.0
- JMS 1.1
- JTA 1.0
- JavaMail 1.3
- JAF 1.0
- JAXP 1.2
- Connector 1.5
- Web Services 1.1
- JAX-RPC 1.1
- SAAJ 1.2
- JAXR 1.0
- J2EE Management 1.0
- JMX 1.2
- J2EE Deployment 1.1
- JACC 1.0

Judd Solutions



## JST Scope



- JSR 45 – Debugging Support
- JSR 175 – Metadata (annotations)
- JSR 181 – Metadata for Web Services
- Integration with popular open source application servers

**Judd Solutions**



## JST Future Scope (2.0)



- EJB 3.0 (Dali)
- JavaServer Faces (JSF)

**Judd Solutions**



## JST Annotations



- Plug-able annotations platform
- XDoclet
  - Open Source
  - Attribute Oriented Programming (AOP) framework
  - [xdoclet.sourceforge.net](http://xdoclet.sourceforge.net)
- Generates
  - Bean Interfaces
  - Bean Util classes
  - EJB Deployment Descriptors
  - Web Deployment Descriptors

**Judd Solutions**




## JST Projects




- Projects Types
  - Enterprise Application Project (ear)
  - EJB Project (ejb-jar)
  - Dynamic Web Project (war)
  - Application Client Project (jar)
  - Connector Project (rar)

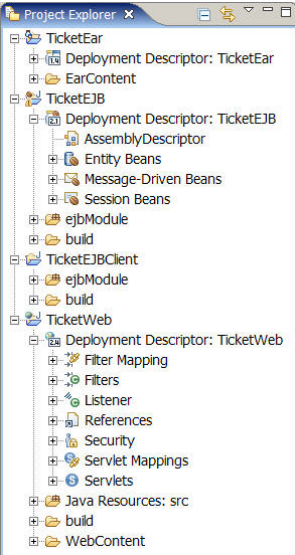
**Judd Solutions**




# JST Projects




- Project Layouts
  - Project per module

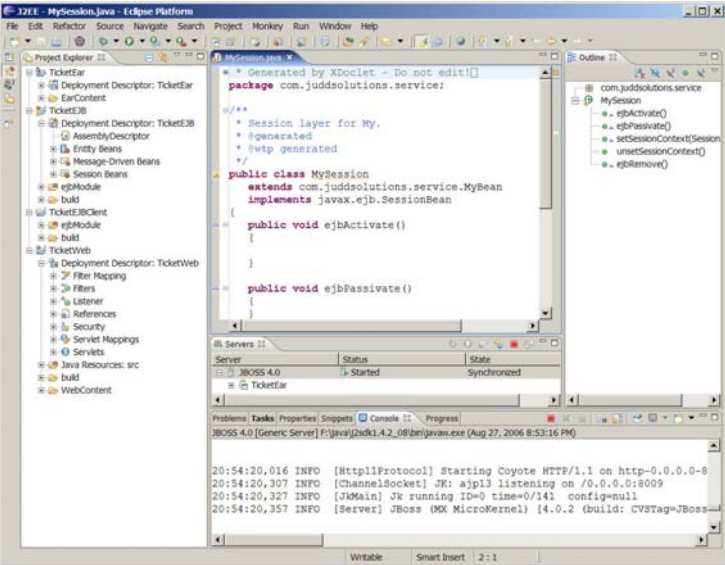


Judd Solutions



# J2EE Perspective





Judd Solutions





## Agenda



- J2EE Landscape
- WTP Overview
- WST Overview
- JST Overview
- EJBs
- Web Artifacts
- Web Services
- Packaging
- Conclusion

**Judd Solutions**



## EJB Types



- Session – server-side business logic
  - Stateful
  - Stateless
- Entity – persistable business objects
  - CMP
  - BMP
- Message-Driven – consumers of JMS messages

**Judd Solutions**



- Session
  - Remote Interface
  - Home Interface
  - Local Interface
  - Local Home Interface
  - Bean Class
  - EJB Deployment Descriptor
  - App Server Specific EJB Deployment Descriptor



- Entity
  - Remote Interface
  - Home Interface
  - Local Interface
  - Local Home Interface
  - Bean Class
  - EJB Deployment Descriptor
  - App Server Specific EJB Deployment Descriptor
  - **App Server Specific CMP Deployment Descriptor**



## EJB Files



- Message-Driven Bean
  - Bean Class
  - EJB Deployment Descriptor
  - App Server Specific EJB Deployment Descriptor

**Judd Solutions**



## Workspace Setup



- Configure application server runtime
- Configure J2EE Annotations

**Judd Solutions**



## Supported Servers



- Apache Tomcat
  - 3.2
  - 4.0
  - 4.1
  - 5.0
  - 5.5
- Apache Geronimo 1.0
- BEA WebLogic 8.1 & 9.0
- IBM WebSphere 6.0.x
- JBoss 3.2.x & 4.0.x
- JOnAS 4.x
- Oracle OC4J Standalone 10.1.3

**Judd Solutions**



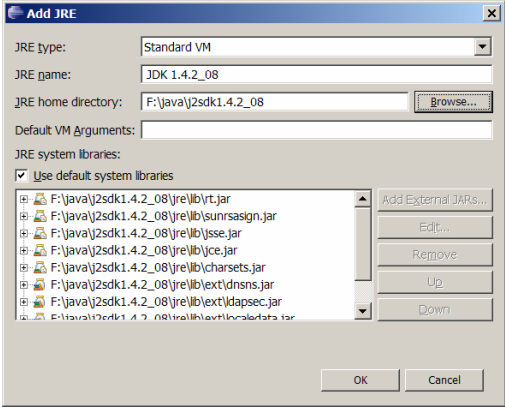
## Defining Server



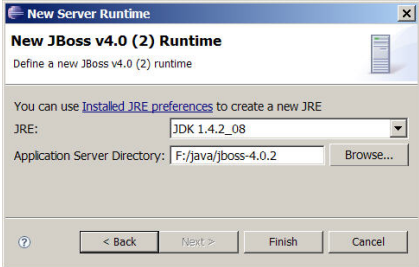
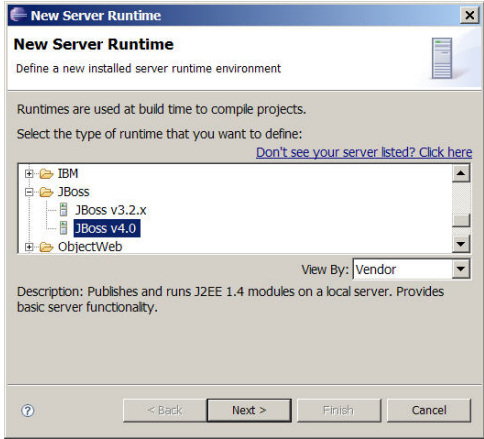
1. Configure server appropriate JDK
2. Configure server runtime
3. Test server

**Judd Solutions**

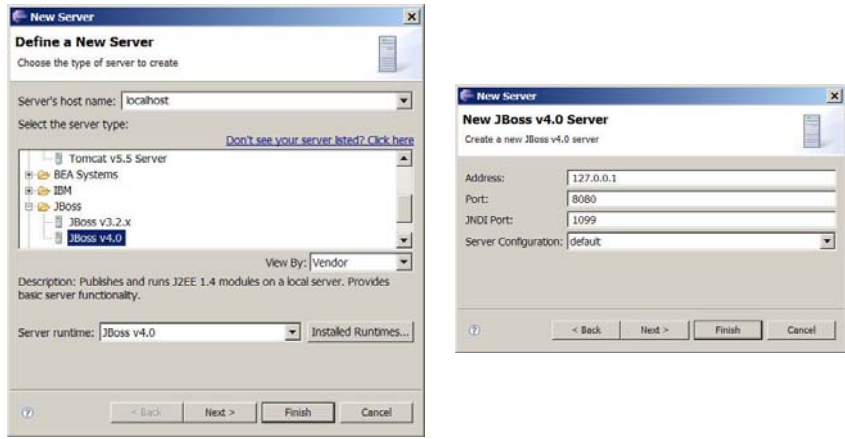
- Window > Preferences > Java > Installed JREs



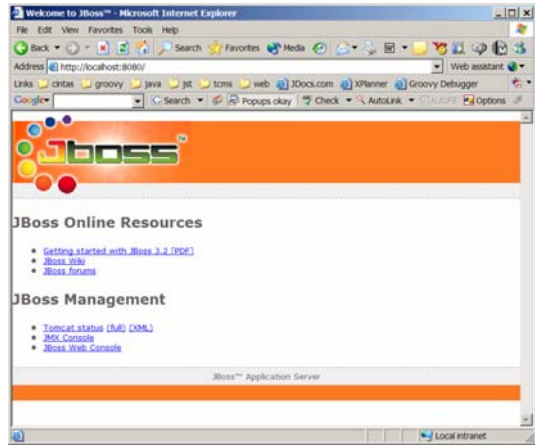
- Window > Preferences > Server > Installed Runtimes



- Use Server View in J2EE Perspective
1. Add server (New > Server)



2. Start server
3. Verify server is up

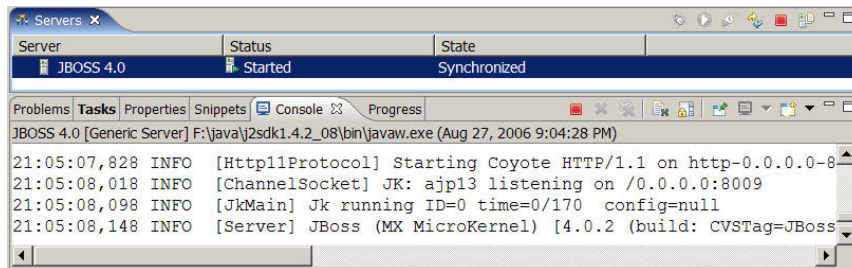




## Server View Usability Tip



- Move Server View above Console view
  - Easier application server startup
  - Easier deployment



Judd Solutions




## Workspace Setup



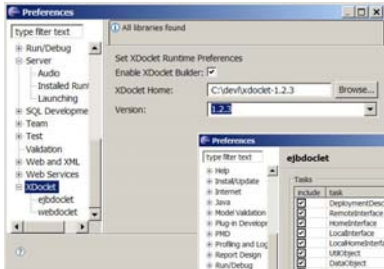
- Configure application server
- Configure J2EE Annotations

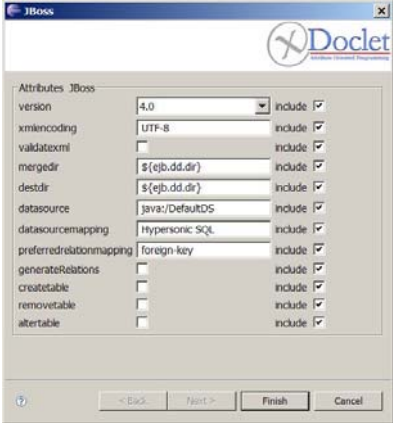
Judd Solutions

## XDoclet Configurations




- Location
- App server & version





Judd Solutions

## Create Session Bean




1. Create EJB, EJB Client & EAR Projects
2. Run XDoclet Enterprise JavaBean Wizard
3. Add XDoclet.ejb annotations
4. Add XDoclet app server specific annotations
5. Deploy

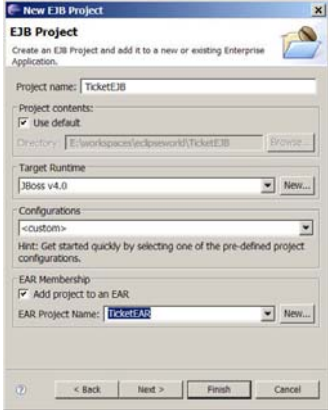
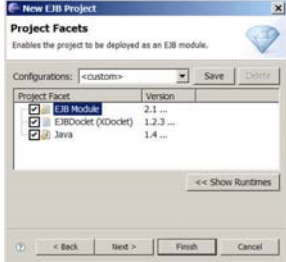

Judd Solutions




## Create Projects




- New > Project > EJB > EJB Project
  - Create EJB, EJB Client and EAR Projects
  - Select Application Server

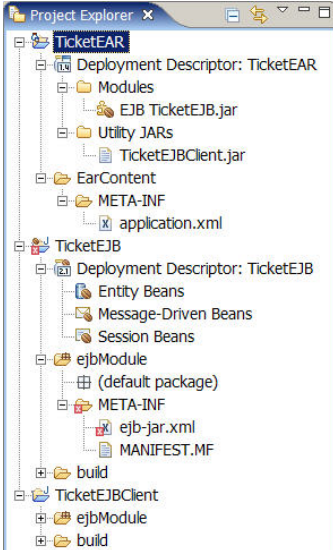







## Create Projects Output



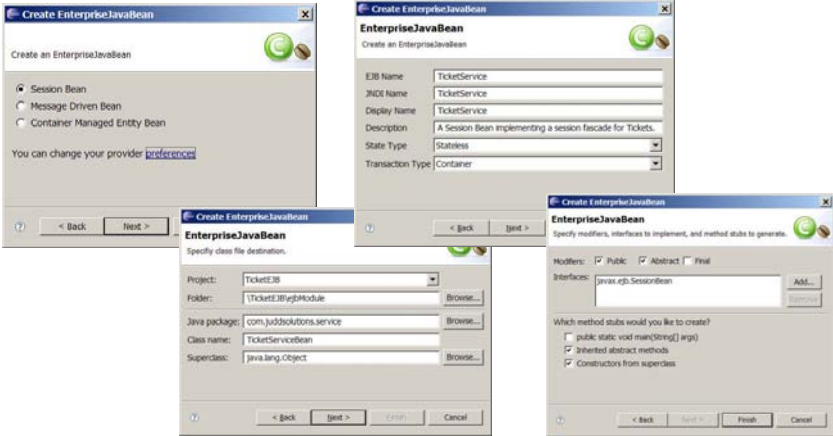
- EJB project
- EJB Client project
- EAR project
  
- Error because there are not any EJBs





# Create Session Bean

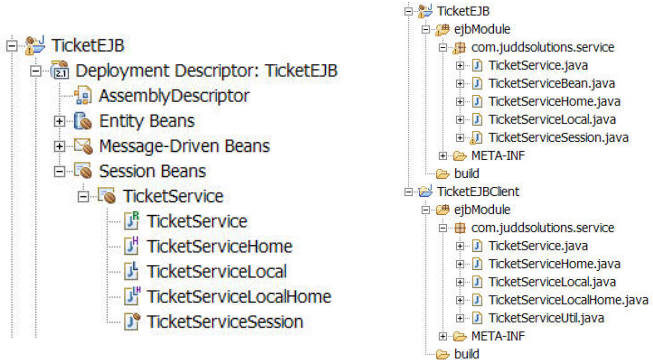
- New > Other > EJB > XDoclet Enterprise JavaBean



Judd Solutions

# Enterprise JavaBean Wizard Output

- \*Bean is the only wizard generated class



Judd Solutions



## Enterprise JavaBean Source



```
package com.juddsolutions.service;

/**
 * <!-- begin-user-doc -->
 * A generated session bean
 * <!-- end-user-doc -->
 */
 * <!-- begin-xdoclet-definition -->
 * @ejb.bean name="TicketService"
 *   description="A Session Bean implementing a session facade for Tickets."
 *   display-name="TicketService"
 *   jndi-name="TicketService"
 *   type="Stateless"
 *   transaction-type="Container"
 * <!-- end-xdoclet-definition -->
 * @generated
 */

public abstract class TicketServiceBean implements javax.ejb.SessionBean {

    /**
     * <!-- begin-xdoclet-definition -->
     * @ejb.interface-method view-type="remote"
     * <!-- end-xdoclet-definition -->
     * @generated
     */
    public String foo(String param) { return null; }
}
```

Judd Solutions



## XDoclet Output



- Generates Home, Remote, EJB Deployment Descriptors, App Server DD and Util
  - Don't need to version control the generated classes
  - Don't modify the generated classes
- Changes to \*Bean causes XDoclet to regenerate

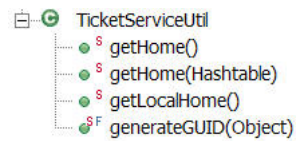
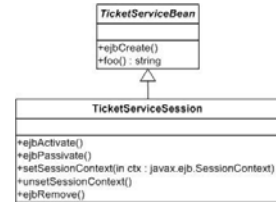
Judd Solutions



## XDoclet Convention



- Bean class
  - Contains business methods
  - Session descendant implements EJB lifecycle methods
- Util class
  - Locator methods
  - GUID generation
- Local Home and Remote Home
  - Component Name
  - JNDI Name



Judd Solutions



## XDoclet annotations



- Code Assist (ctrl + space)

```

/**
 * <!-- begin-xdoclet-definition -->
 * @ejb.interface-method view-type="both"
 * <!-- end-xdoclet-definition -->
 * @generated
 */
public String foo(String param) {
    return param + " " + new Date();
}
  
```

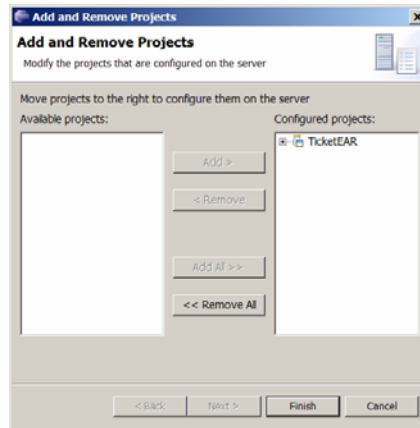
Judd Solutions



## Deploy



- Right click on server and Add and Remove Projects



Judd Solutions



## Create Session Bean Client

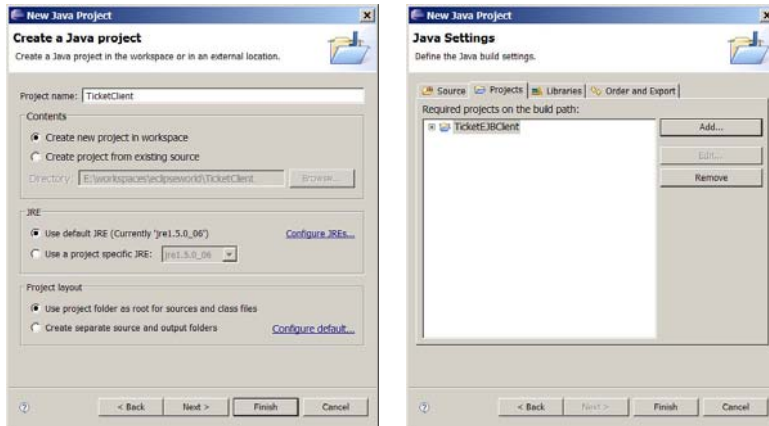


1. Create Java project
2. Create client class
3. Add app server client jars
4. Create jndi.properties file
5. Run

Judd Solutions

## Create Java Project

- New > Project > Java > Java Project
- Add required \*EJBClient project



Judd Solutions

## Create Client Class

- New > Class
- Add main method



Judd Solutions



## Client Source



- Mostly boiler plate EJB client code

```
public class RemoteClient {  
  
    /**  
     * Main entry point to client  
     * @param args command line args  
     * @throws Exception any possible exception  
     */  
    public static void main(String[] args) throws Exception {  
        TicketServiceHome home = TicketServiceUtil.getHome();  
        TicketService service = home.create();  
        String result = service.foo("cool");  
        System.out.println(result);  
        System.out.println("\ndone");  
    }  
}
```

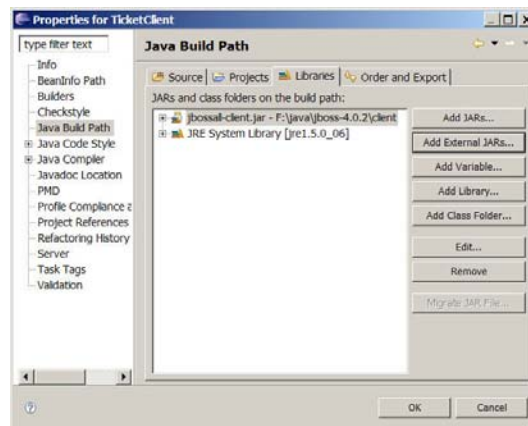
Judd Solutions



## App Server Client jars



- Project > Properties > Java Build Path > Libraries



Judd Solutions



## jndi.properties file



- New > File

```
# JBoss specific JNDI properties

java.naming.factory.initial=org.jnp.interfaces.NamingContextFactory
java.naming.provider.url=jnp://localhost:1099
java.naming.factory.url.pkgs=org.jboss.naming:org.jnp.interfaces
```

**Judd Solutions**



## Create Entity Bean



1. Run XDoclet Enterprise JavaBean Wizard
2. Add XDoclet ejb annotations
  - Map properties to table columns
3. Add XDoclet app server specific annotations
4. Add getData() method
5. Deploy

**Judd Solutions**



- New > Other > EJB > XDoclet Enterprise JavaBean

The screenshots show the following steps in the wizard:

- Step 1:** Selecting 'Container Managed Entity Bean'.
- Step 2:** Project: TicketJ2E, Folder: TicketJ2EModule, Java package: com.juddsolutions.entity, Class name: TicketBean, Superclass: java.lang.Object.
- Step 3:** EJB Name: Ticket, Schema: Ticket, Display Name: Ticket, Description: An EJB named Ticket, CMP Version: 2.4, Unpacker: Define new attributes.
- Step 4:** Modifiers: Public, Abstract, Final. Interface: java.sql.EntityBean. Method stubs: public static void main(String[] args), @Override abstract methods, Constructors from superclass.
- Step 5:** Table: TICKET. Managed Attributes table:
 

Name	Column	Type	JDBC Type	SQL Type	Read Only	Primary
id	ID	Integer	BIGINT	BIGINT	<input type="checkbox"/>	<input checked="" type="checkbox"/>
summary	SUMMARY	java.lang.String	VARCHAR	VARCHAR	<input type="checkbox"/>	<input type="checkbox"/>
detail	DETAIL	java.lang.String	VARCHAR	VARCHAR	<input type="checkbox"/>	<input type="checkbox"/>
email	EMAIL	java.lang.String	VARCHAR	VARCHAR	<input type="checkbox"/>	<input type="checkbox"/>
submitted	SUBMITTED	java.sql.Timestamp	TIMESTAMP	TIMESTAMP	<input type="checkbox"/>	<input type="checkbox"/>
lastModified	LASTMODIFIED	java.sql.Timestamp	TIMESTAMP	TIMESTAMP	<input type="checkbox"/>	<input type="checkbox"/>

- \*Bean is the only wizard generated class

The screenshots show the following project structures:

- Left Screenshot (TicketEJB):**
  - Deployment Descriptor: TicketEJB
    - AssemblyDescriptor
      - Entity Beans
        - Ticket
          - detail : java.lang.String
          - email : java.lang.String
          - id : java.lang.Integer
          - Integer
          - lastModified : java.sql.Timestamp
          - Query SELECT OBJECT(a) FROM ticket as a
          - findAll
          - submitted : java.sql.Timestamp
          - summary : java.lang.String
          - TicketCMP
          - TicketLocal
          - TicketLocalHome
    - Message-Driven Beans
    - Session Beans

- Right Screenshot (TicketEJBClient):**
- ejbModule
  - com.juddsolutions.entity
    - TicketBean.java
    - TicketCMP.java
  - com.juddsolutions.service
    - META-INF
  - build
  - TicketEJBClient
    - ejbModule
      - com.juddsolutions.entity
        - Ticket.java
        - TicketData.java
        - TicketHome.java
        - TicketLocal.java
        - TicketLocalHome.java
        - TicketUtil.java
      - com.juddsolutions.service
        - META-INF



## Entity Bean Source (Class)



```

package com.juddsolutions.entity;

/**
 * @ejb.bean name="Ticket"
 *   jndi-name="ticket"
 *   type="CMP"
 *   primaryKey-field="id"
 *   schema="ticket"
 *   cmp-version="2.x"
 *   view-type="local"
 * @ejb.persistence
 *   table-name="TICKET"
 * @ejb.finder
 *   query="SELECT OBJECT(a) FROM ticket as a"
 *   signature="java.util.Collection findAll()"
 * @ejb.pk class="Integer"
 * @jboss.persistence datasource="java:/jdbc/ticket"
 *   datasource-mapping="Derby"
 *   table-name="TICKET"
 *   create-table="false"
 *   remove-table="false"
 *   alter-table="false"
 * @jboss.unknown-pk class = "java.lang.Integer"
 *   readonly = "true"
 *   column-name = "ID"
 *   auto-increment = "true"
 * @jboss.entity-command name = "get-generated-keys"
 *   class = "org.jboss.ejb.plugins.cmp.jdbc.keygen.JDBC30GeneratedKeysCreateCommand"
 */
public abstract class TicketBean implements javax.ejb.EntityBean {
    // excluded details
}

```

Judd Solutions



## Entity Bean Source (Method)



```

/**
 * @ejb.create-method
 */
public Integer ejbCreate() throws javax.ejb.CreateException {
    return null;
}

/**
 * @ejb.persistent-field
 * @ejb.persistence
 *   column-name="ID"
 *   jdbc-type="BIGINT"
 *   sql-type="BIGINT"
 *   read-only="false"
 * @ejb.pk-field
 * @ejb.interface-method
 */
public abstract Integer getId();

/**
 * @ejb.interface-method
 */
public abstract void setId(Integer id);

/**
 * @ejb.persistent-field
 * @ejb.persistence
 *   column-name="SUMMARY"
 *   jdbc-type="VARCHAR"
 *   sql-type="VARCHAR"
 *   read-only="false"
 * @ejb.interface-method
 */
public abstract java.lang.String getSummary();

/**
 * @ejb.interface-method
 */
public abstract void setSummary(java.lang.String summary);

```

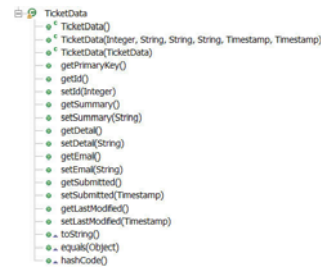
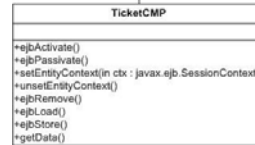
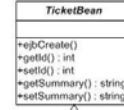
Judd Solutions



## XDoclet Convention



- Bean class
  - Contains entity properties
  - Entity descendant implements EJB lifecycle methods
- Util class
  - Locator methods
  - GUID generation
- Local Home
  - Component Name
  - JNDI Name
- Data Transfer Object (DTO)



Judd Solutions



## Convert Entity Bean to DTO



- Add getData() method to \*Bean class
  - Implemented by \*CMP
- Generated \*Data class

```

/**
 * Provides access to the generated getData() method of the generated
 * CMP class.
 * @ejb.interface-method
 */
public abstract TicketData getData();
  
```

Judd Solutions



## Session Façade Implementation



- Added to TicketServiceBean

```
/**
 * Returns all tickets.
 * @ejb.interface-method view-type="both"
 */
public Collection retrieveAllTickets() throws FinderException,
    NamingException {
    Collection ticketDTOs = new ArrayList();
    Collection tickets = TicketUtil.getLocalHome().findAll();
    for (Iterator iter = tickets.iterator(); iter.hasNext();) {
        TicketLocal ticket = (TicketLocal) iter.next();
        ticketDTOs.add(ticket.getData());
    }
    return ticketDTOs;
}
```

Judd Solutions



## Agenda



- J2EE Landscape
- WTP Overview
- WST Overview
- JST Overview
- EJBs
- Web Artifacts
- Web Services
- Packaging
- Conclusion

Judd Solutions



## Setup Web Project



1. Create Dynamic Web Project
2. Add web framework libraries and configurations
3. Create web artifacts
  - JavaServer Pages
  - Servlets
  - Struts Action
  - Etc.

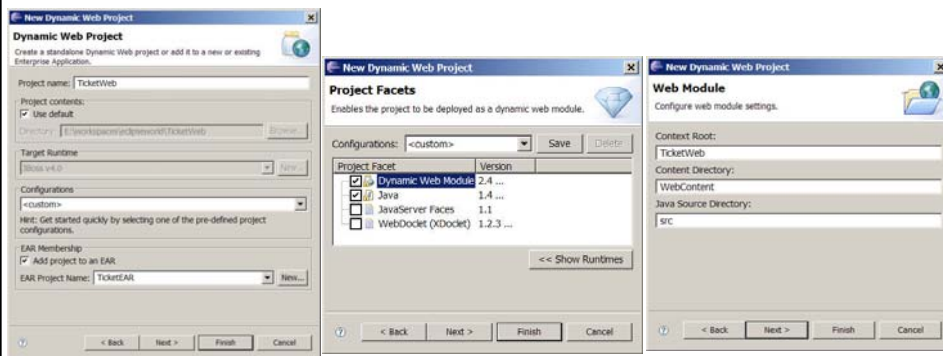
Judd Solutions



## Create Web Project



- File > New > Project > Dynamic Web Project



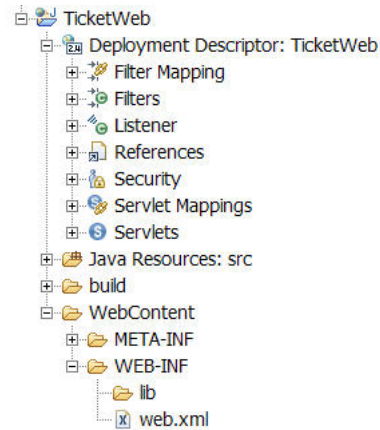
Judd Solutions



## Web Project Output



- WebContent is the root context
- Web application deployment descriptor
  - web.xml



Judd Solutions



## Setup JSTL & Struts



1. Add Struts and JSTL jars to the WEB-INF/lib directory
2. Configure ActionServlet in web.xml
3. Configure Struts and JSTL tag libraries
  - Copy tag library definition files
  - Map taglib
4. Create Struts config file

Judd Solutions



## web.xml



```

<?xml version="1.0" encoding="UTF-8"?>
<web-app id="WebApp_ID" version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
  <display-name>
    TicketWeb</display-name>

  <servlet>
    <servlet-name>action</servlet-name>
    <servlet-class>org.apache.struts.action.ActionServlet</servlet-class>
    <init-param>
      <param-name>config</param-name>
      <param-value>/WEB-INF/struts-config.xml</param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
  </servlet>

  <servlet-mapping>
    <servlet-name>action</servlet-name>
    <url-pattern>*.do</url-pattern>
  </servlet-mapping>

  <jsp-config>
    <taglib>
      <taglib-uri>struts-bean</taglib-uri>
      <taglib-location>/WEB-INF/struts-bean.tld</taglib-location>
    </taglib>
    <taglib>
      <taglib-uri>struts-html</taglib-uri>
      <taglib-location>/WEB-INF/struts-html.tld</taglib-location>
    </taglib>
    <taglib>
      <taglib-uri>jstl-core</taglib-uri>
      <taglib-location>/WEB-INF/c.tld</taglib-location>
    </taglib>
  </jsp-config>
</web-app>

```

Judd Solutions



## struts-config.xml



```


<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE struts-config PUBLIC "-//Apache Software
Foundation//DTD Struts Configuration 1.1//EN"
"http://jakarta.apache.org/struts/dtds/struts-config_1_1.dtd">

<struts-config>
  <form-beans>
  </form-beans>
  <global-exceptions>
  </global-exceptions>
  <global-forwards>
  </global-forwards>
  <action-mappings>
  </action-mappings>
</struts-config>

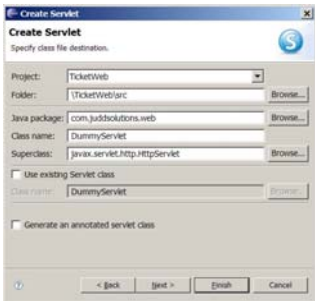
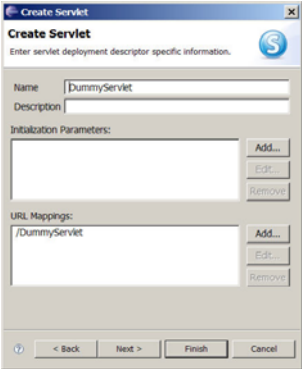

```

Judd Solutions

## Create Servlet




- File > New > Other > Web > Servlet






**Judd Solutions**

## Create Struts Action



- File > New > Other > Class



**Judd Solutions**





## Struts Action Implementation



```
package com.juddsolutions.web;

import java.util.Collection;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;

import com.juddsolutions.service.TicketServiceLocal;
import com.juddsolutions.service.TicketServiceUtil;

public class ListAllTickets extends Action {

    private static final String SUCCESS_FORWARD = "success";

    public ActionForward execute(ActionMapping mapping, ActionForm form,
        HttpServletRequest request, HttpServletResponse response) throws Exception
    {
        TicketServiceLocal service = TicketServiceUtil.getLocalHome().create();
        Collection tickets = service.retrieveAllTickets();
        request.setAttribute("tickets", tickets);
        return mapping.findForward(SUCCESS_FORWARD);
    }
}
```

Judd Solutions




## Struts Action Mapping




```
<action-mappings>
  <action
    type="com.juddsolutions.web.action.ListAllTicketsAction"
    scope="request" path="/index">
    <forward name="success" path="/WEB-INF/jsp/listtickets.jsp" />
  </action>
</action-mappings>
```

Judd Solutions



# Create JSP



- File > New > Other > Web > JSP

**New JavaServer Page**

JavaServer Page  
Create a new JavaServer Page.

Enter or select the parent folder:

TicketWeb/WebContent/WEB-INF/jsp

- TicketEJB
- TicketEJBClient
- TicketWeb
  - settings
  - build
  - src
  - WebContent
    - META-INF
    - WEB-INF
      - JSP
      - lib

File name: listtickets.jsp

Advanced >>

**New JavaServer Page**

Select JSP Template

Select a template as initial content in the JSP page.

Use JSP Template


Templates are 'New JSP' templates found in the JSP Templates preference page.

Name	Description
New JSP File (html)	JSP with html markup
New JSP File (xhtml)	JSP with xhtml markup
New JSP File (xhtml, xml syntax)	JSP with xhtml markup and xml style s...


Preview

```
<%@ page language="java" contentType="text/html; charset=${encoding}"
pageEncoding="${encoding}" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional/EN" >
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=${encoding}" >
<title>Insert title here</title>
</head>
</body>
</html>
```

Judd Solutions



# listtickets.jsp



```
<html>
<head>
<link rel="stylesheet" type="text/css" href="styles/default.css"/>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<%@ taglib uri="jstl-core" prefix="c" %>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Trouble Ticket Application</title>
</head>
<body>
<h2>Trouble Tickets</h2>
<table width="100%" border="0">
<tr class="add">
<td><a href="/newticket.do"> Add Ticket</a></td>
</tr>
</table>
<table width="100%">
<tr class="heading">
<th>ID</th>
<th>Summary</th>
<th>Submitted Date</th>
</tr>
</table>
<c:forEach items="${tickets}" var="ticket">
<tr class="row">
<td>&nbsp;&nbsp;&nbsp;<c:out value="${ticket.id}"/></td>
<td>&nbsp;&nbsp;&nbsp;<c:out value="${ticket.summary}"/></td>
<td>&nbsp;&nbsp;&nbsp;<c:out value="${ticket.submitted}"/></td>
</tr>
</c:forEach>
</table>
<table width="100%" border="0">
<tr class="add">
<td><a href="/newticket.do"> Add Ticket</a></td>
</tr>
</table>
</body>
</html>
```

Judd Solutions



## Web Application



### Trouble Tickets

[+ Add Ticket](#)

ID	Summary	Submitted Date
1	Not enough memory.	2005-09-11 20:36:29.556
2	Computer wont.	2005-09-11 20:36:29.626
3	Windows Crashed Again	2005-09-11 20:36:29.676
4	Popups are slowing down my computer	2005-09-11 20:36:29.716
5	Monitor flickers cool.	2005-09-11 20:36:29.746
6	Cannot connect to the network.	2005-09-11 20:36:29.786
7	Not enough memory.	2005-09-11 20:36:29.816
9	new ticket	2005-09-11 20:50:16.615
10	ticket1	2006-05-09 13:14:32.17
11	ticket2	2006-05-09 13:14:42.595
12	new ticket	2006-05-16 20:30:06.278

[+ Add Ticket](#)

Judd Solutions



## Agenda



- J2EE Landscape
- WTP Overview
- WST Overview
- JST Overview
- EJBs
- Web Artifacts
- Web Services
- Packaging
- Conclusion

Judd Solutions



## 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

Judd Solutions




## Apache Axis




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

Judd Solutions



# 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](http://www.xmethods.net))
- Service description
  - [www.webservicex.net/WeatherForecast.asmx](http://www.webservicex.net/WeatherForecast.asmx)


Zip code:

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 Forecast:

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

**Judd Solutions**



# Weather Forecast Web Service



## Service description

WeatherForecast

WeatherForecastSoap

<http://www.webser...>

→

WeatherForecastSoap

GetWeatherByZipCode

Input parameters: GetWeatherByZipCode

Output parameters: GetWeatherByZipCodeResponse

GetWeatherByPlaceName

Input parameters: GetWeatherByPlaceName

Output parameters: GetWeatherByPlaceNameResponse

## Data transfer objects

WeatherForecasts

- Latitude [1..1] float
- Longitude [1..1] float
- AllocationFactor [1..1] float
- FipsCode [0..1] string
- PlaceName [0..1] string
- StateCode [0..1] string
- Status [0..1] string
- Details [0..1] ArrayOfWeatherData

→

ArrayOfWeatherData

WeatherData [0..\*] WeatherData

→

WeatherData

- Day [0..1] string
- WeatherImage [0..1] string
- MaxTemperatureF [0..1] string
- MinTemperatureF [0..1] string
- MaxTemperatureC [0..1] string
- MinTemperatureC [0..1] string

**Judd Solutions**



## 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

**Judd Solutions**



## Consuming Web Services Steps



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

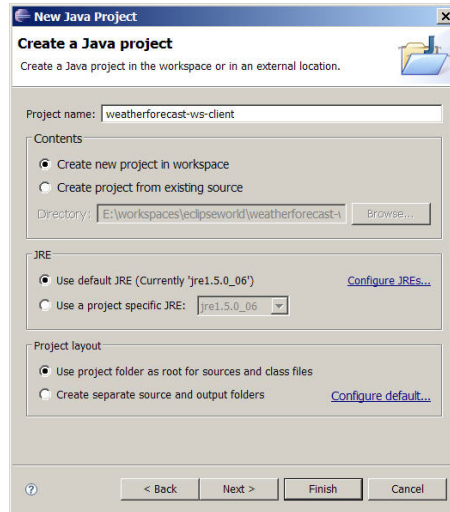
**Judd Solutions**



## Create Project



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



Judd Solutions



## 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

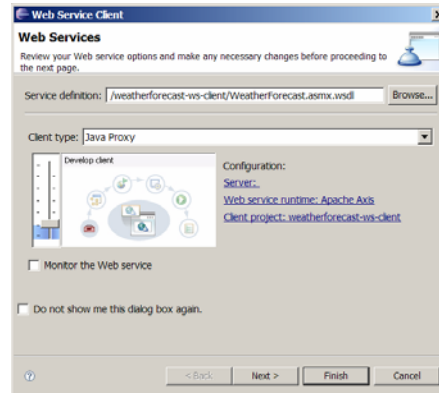
Judd Solutions



## 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 Monitor



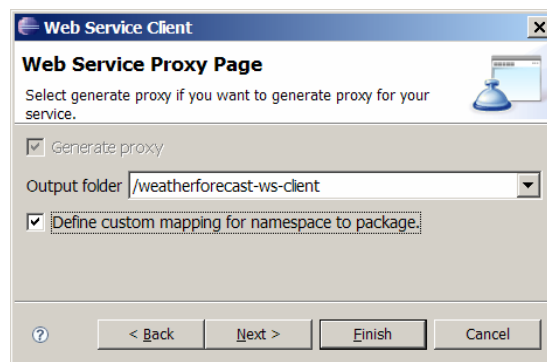
Judd Solutions



## Generate Web Service Client



- Specify project to generate the code in



Judd Solutions

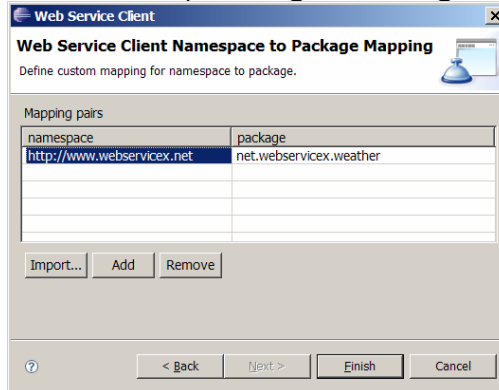




## Generate Web Service Client



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



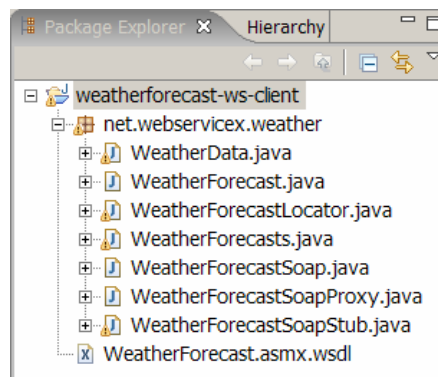
Judd Solutions



## 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.

Judd Solutions



## Using Generate Classes



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

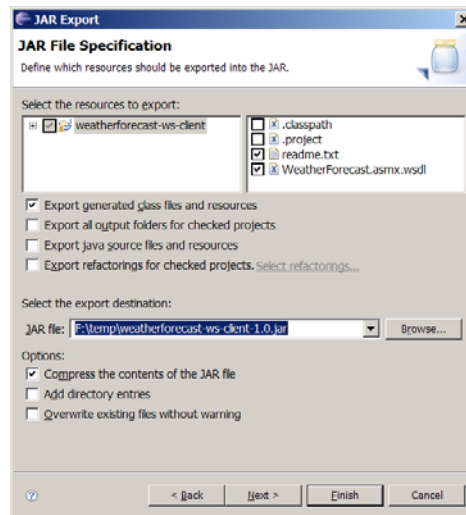
System.out.println(forecasts.getLatitude());
System.out.println(forecasts.getLongitude());

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());
}
```

Judd Solutions



## Export client Jar



Judd Solutions



## Producing Web Services



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

**Judd Solutions**




## Bottom Up Steps



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

**Judd Solutions**

## Create Dynamic Web Project



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

**New Dynamic Web Project**

**Dynamic Web Project**  
Create a standalone Dynamic Web project or add it to a new or existing Enterprise Application.

Project name: | order-ws-server

Project contents:  
 Use default  
 Directory: | E:\workspaces\ eclipseworld\order-ws-server | Browse...


Target Runtime:  
 Apache Tomcat v5.5 | New...

Configurations:  
 <custom> |  
 Hint: Get started quickly by selecting one of the pre-defined project configurations.

EAR Membership  
 Add project to an EAR.  
 EAR Project Name: | order-ws-serverEAR | New...

Judd Solutions

## Create Dynamic Web Project



- Facets and versions

**New Dynamic Web Project**

**Project Facets**  
Adds support for writing applications using Java programming language.

Configurations: | <custom> | Save | Delete

Project Facet	Version
<input checked="" type="checkbox"/> Dynamic Web Module	2.4 ...
<input checked="" type="checkbox"/> Java	5.0 ...
<input type="checkbox"/> JavaServer Faces	1.1
<input type="checkbox"/> WebDoclet (XDoclet)	1.2.3 ...

Judd Solutions



## Create Dynamic Web Project



- Context Root
- Directories

**New Dynamic Web Project**

**Web Module**  
Configure web module settings.

Context Root:  
order-ws-server

Content Directory:  
WebContent

Java Source Directory:  
src

< Back   Next >   Finish   Cancel

Judd Solutions




## Create Service



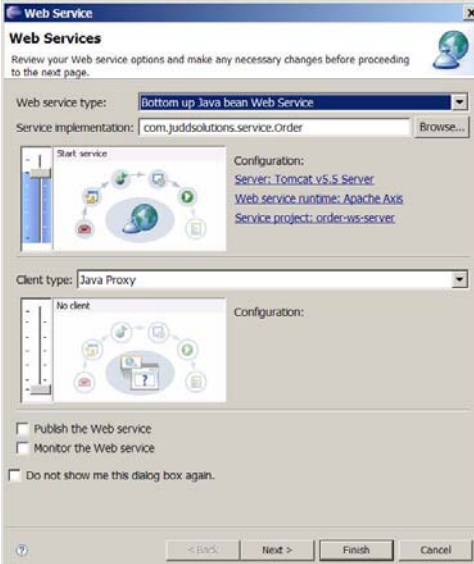
```
public class Order {  
  
    private static final float DISCOUNT = 0.9f;  
  
    public float quote(float price, int quantity) {  
        if (quantity > 100) {  
            price = price * DISCOUNT;  
        }  
        return price * quantity;  
    }  
}
```

Judd Solutions

## 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

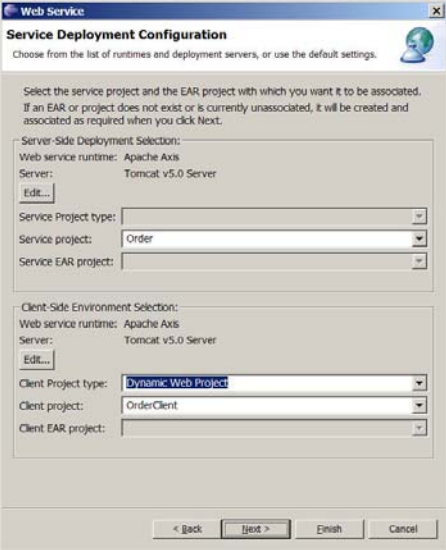


Judd Solutions

## Generate Web Service



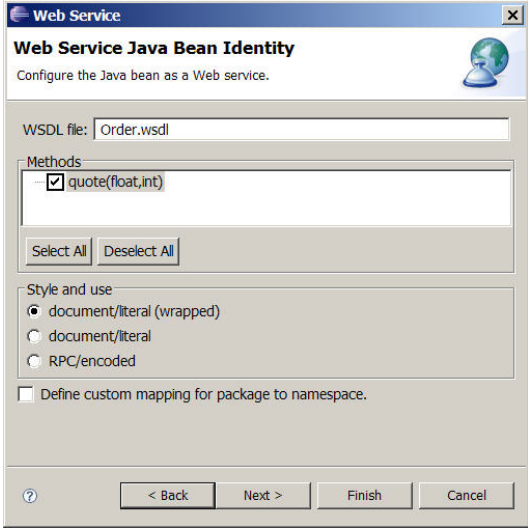
- Select project for deployment descriptors
- Create new client project



Judd Solutions

**Generate Web Service**

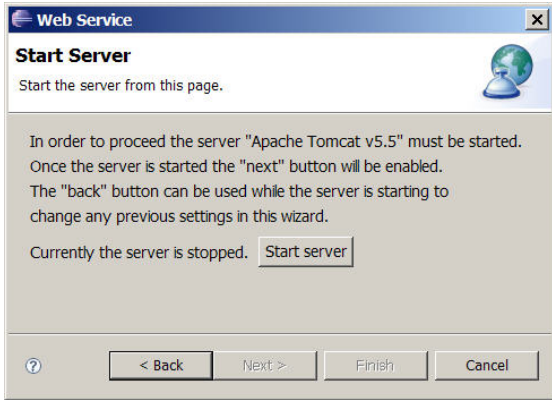
- Select methods to expose
- Select type
  - RPC
    - Performance
    - Simple types
  - Document
    - Interoperability
    - XML



**Judd Solutions**

**Generate Web Service**

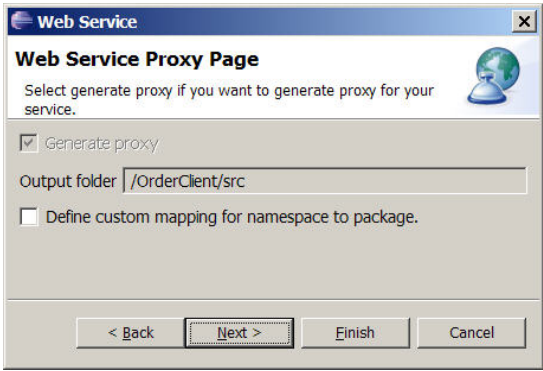
- Start server to host web service and test client



**Judd Solutions**

## Generate Web Service

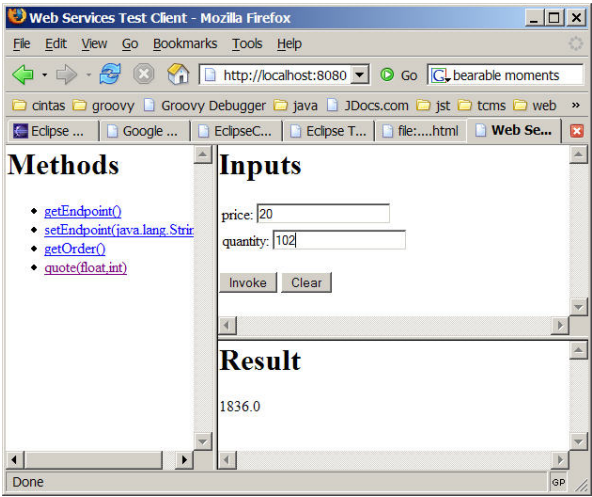
- Directory for client code
- Optionally, provide namespace to package mapping



**Judd Solutions**

## Generate Web Service

- Test Client



**Judd Solutions**

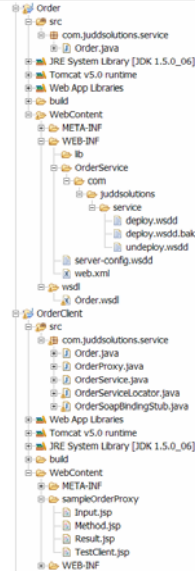




## Output



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



Judd Solutions




## Data Types




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

Judd Solutions




# Agenda




- J2EE Landscape
- WTP Overview
- WST Overview
- JST Overview
- EJBs
- Web Artifacts
- Web Services
- **Packaging**
- Conclusion

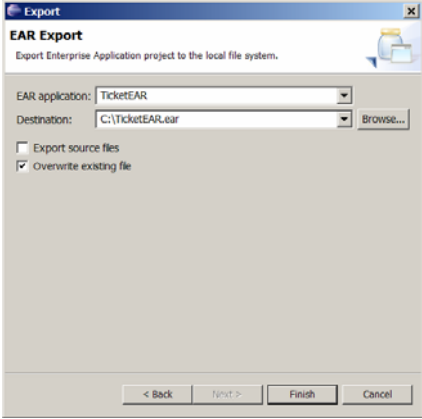
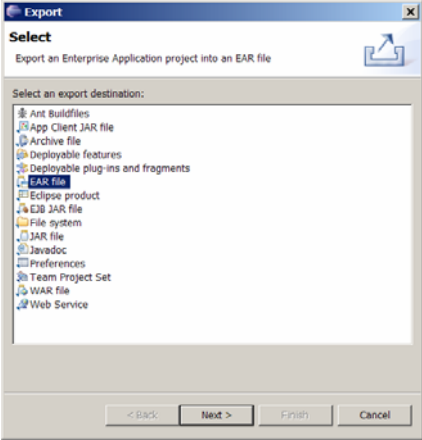
**Judd Solutions**



# Packaging



- Create a deployable version
- **Export > EAR file**



**Judd Solutions**



## Agenda



- J2EE Landscape
- WTP Overview
- WST Overview
- JST Overview
- EJBs
- Web Artifacts
- Web Services
- Packaging
- Conclusion

**Judd Solutions**



## Conclusion



JST provides the facilities necessary for developing, deploying and testing J2EE applications using XDoclet for annotations and code generation.

**Judd Solutions**



## WTP Resources



- [www.eclipse.org/webtools/](http://www.eclipse.org/webtools/)
- [www.projst.com](http://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>
- Dali
  - <http://www.eclipse.org/dali/>
- XDoclet
  - <http://xdoclet.sourceforge.net>

Judd Solutions



## J2EE Resources



- [www.javasoft.com](http://www.javasoft.com)
- [www.theserverside.com](http://www.theserverside.com)
- J2EE 1.4 Specification
  - [java.sun.com/j2ee/j2ee-1\\_4-fr-spec.pdf](http://java.sun.com/j2ee/j2ee-1_4-fr-spec.pdf)
  - [java.sun.com/j2ee/1.4/docs/api/](http://java.sun.com/j2ee/1.4/docs/api/)
- EJB 2.1 Specification
  - [java.sun.com/products/ejb/docs.html](http://java.sun.com/products/ejb/docs.html)

Judd Solutions



## Contact Information



- <http://www.juddsolutions.com>
- [cjudd@juddsolutions.com](mailto: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>



Judd Solutions



## Questions ?

Please fill out your evaluations.

Slides can be found at

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

Judd Solutions