

Omaha WebSphere User Group Meeting

Presented by:

Sheng Sheen

Senior Consultant

Capstone Consulting, Inc.

Part II: Jakarta Struts



CAPSTONE

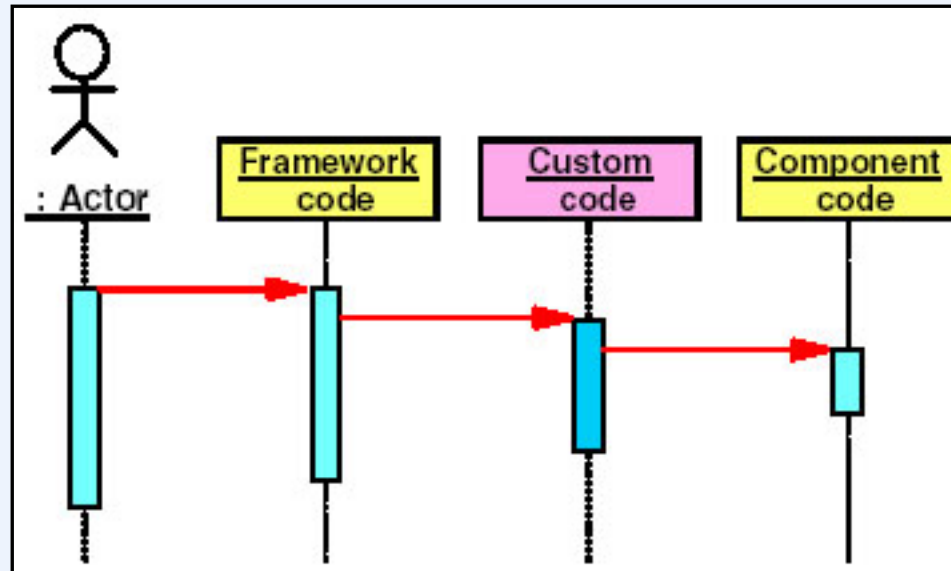
Building **Solid** Enterprise **Solutions**

Struts

- What is Struts?
 - Framework that helps developers build web applications quickly and easily
- Struts relies on standard Java and J2EE technologies:
 - JavaBeans
 - Java Servlets
 - JavaServer Pages (JSPs)

What is a Framework?

- a framework is a software piece that an application development can start with; it is a basis



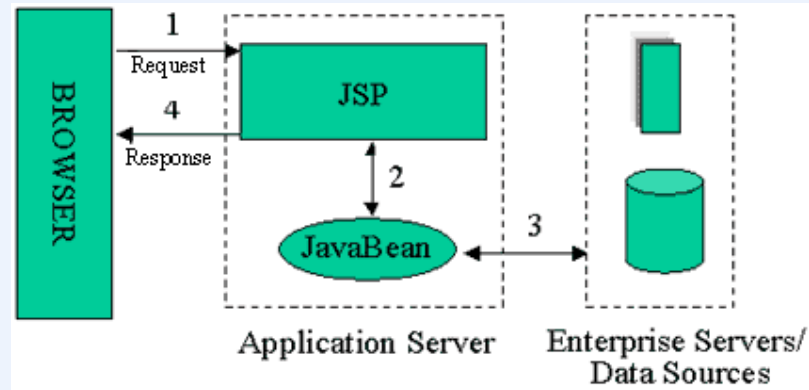
History of Struts

- Open Source
- Part of the Apache Jakarta Project
- <http://jakarta.apache.org/struts>
- Creator: Craig R. McClanahan – also the primary architect of Tomcat 4, and implementation architect of Java Web Services Developer Pack
- Introduced in May of 2000
- First official release on June, 2001 – Struts version 1.0
- Latest available release Struts v1.1-b3, December 30, 2002

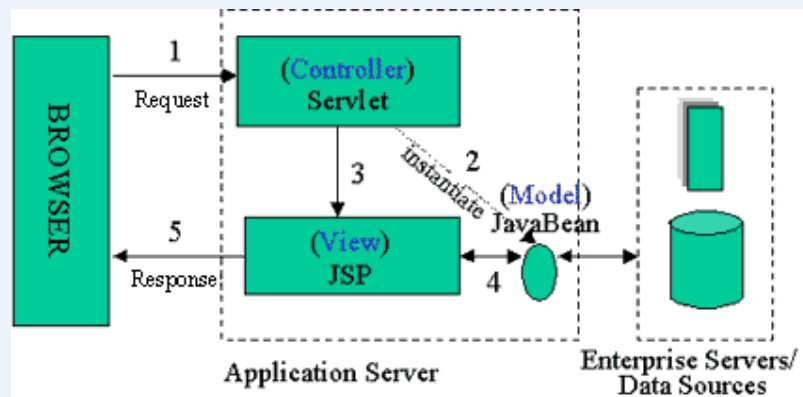
- Goal of the Struts project is to provide an open source framework useful in building Web Applications with Java Servlets and Java Server Pages
- Struts encourages application architectures based on the model-view-controller (MVC) design, also known as Model II of MVC

MVC Pattern

- Model 1 Architecture



- Model 2 Architecture



Model-View-Controller

- Most application designs follow certain documented patterns that have been proven in many successful installations
- Model-view-controller (MVC) pattern:
 - Model : set of objects that represents the business logic of the application
 - View : way of presenting a set of information to a user, particular web page or screen
 - Controller : layer in an application that handles the details of application flow and navigation

J2EE & MVC Pattern

- Mapping for J2EE APIs to the three roles in the MVC pattern:
 - Model – JavaBeans and Enterprise JavaBeans
 - View – Java Server Pages
 - Controller – Servlets
- Servlets act as controllers and are the recipients of HTTP POST requests, and are responsible for passing POSTed data to the model and selecting which JSP page will be invoked to display results

Common Problems in Web Applications

- Mapping HTTP parameters to JavaBean – one of the most common tasks is to map a set of HTTP parameters from command line or from the HTML POST to a JavaBean, it is cumbersome
- Validation – No standard way in servlet/JSP programming to validate that an HTML form is filled in correctly (Note: this is in addition to JavaScript Input Validation)
- Error Messages – No standard way to display error messages in a JSP or Servlet

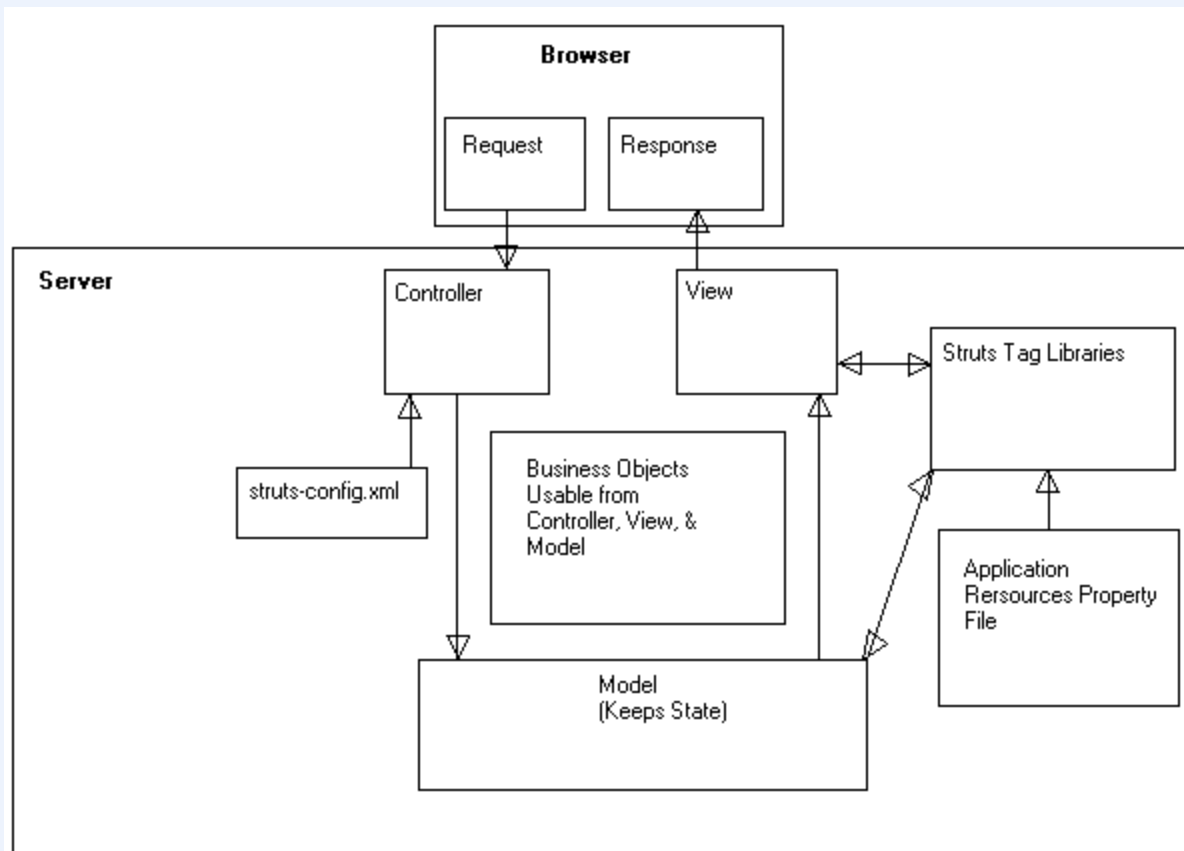
Common Problems in Web Applications

- Message Internationalization – there may be hard-coded messages in the form of short error or information messages
- Hard Coded JSP URIs – JSP pages are usually coded directly into the code of the calling servlet, this makes it impossible to reorganize or rename the JSP without changing the source code in the servlets

- True to the model-view-controller design pattern, Struts applications have three major components:
 - Servlet controller
 - Java Server Pages (the “view”)
 - Application’s business logic (the “model”)

- Struts includes the following functionalities:
 - Servlet controller (ActionServlet) that dispatches requests to appropriate Action classes provided by the developer
 - JSP custom tag libraries, and associated support in the controller servlet, to assist in creation of form-based applications
 - Utility classes to support XML parsing, automation population of JavaBean properties based on the Java Reflection API, and internationalization of prompts and messages

Jakarta Struts



Servlet Controller

- Controller
 1. Intercepts and routes HTTP requests from the client to framework objects and corresponding extended objects
 2. Calls the appropriate user-written Struts Action to perform the task, then forwards to an appropriate view component (JSP) for the response

Servlet Controller

- Primary component of the controller is a servlet of class ***org.apache.struts.action.ActionServlet***
- You will not have to write this servlet
- Initialize the servlet in the web.xml deployment descriptor
- Controller parses a configuration XML file which defines, among other things, the action mappings for the application
- At a minimum, a mapping must specify:
 - Request path
 - Object type to act upon the request
- Struts uses the JAXP API for its XML parsing

Struts Action

- Action objects are linked to the controller, and have access to that servlet's methods
- Action object can handle the request, or indicate the control to be forwarded to another action
- Action objects can indirectly forward one or more shared objects including JavaBeans by placing them in one of the standard collections shared by Java servlets

- The goal of an Action class is to process the request, and return an ActionForward object that identifies the JSP to forward to, so the response can be generated back to the user

- Action Classes:
 - The Action class defines a perform method that you override
 - `public ActionForward perform(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response) throws IOException, ServletException;`

- A typical Action perform() method:
 - Validate user's session information
 - If validation has not yet occurred, validate the form bean properties as necessary
 - Perform the processing required to deal with this request (Call other Java classes or EJBs)
 - Update the server-side objects that will be used to create the next page
 - Return an ActionForward object that identifies the JSP to forward to

Struts Form Bean

- With Struts, the data for an input form can be easily stored in a form bean (ActionForm)
- Form bean is saved in one of the standard, shared context collections, so it can be used by other objects
- Action object receives it as input to perform its tasks

Struts Form Bean

- Extends the Struts ***org.apache.struts.action.ActionForm*** class
- Form Bean will have a HTML form associated to it
- With Form Beans defined in the ActionMapping configuration file, the Servlet Controller will perform the following:
 - Check session for instance of form bean
 - If form bean does not exist, one is created automatically
 - For every input parameter whose name corresponds to the name of a property in the bean, the corresponding setter method will be called
 - The updated form bean will be passed to the Action Class `perform()` method when it is called

Struts Form Bean

- Form bean can be used:
 - To collect data from the user (setters)
 - To validate what the user entered
 - By the JSP to populate the fields (getters)
- ***ActionForm.validate*** method is used to perform any validation, and returns a set of ***ActionError*** objects for failures
- Error messages are held in a properties file that Struts refers to

Struts JSP Custom Tags

- Response back to the users are displayed in JSPs
- Four JSP tag libraries that Struts provides:
 - **HTML** : which includes tags for dynamic pages, especially forms
 - **Beans** : which includes tags for improved access to Java Beans
 - **Logic** : which includes tags that support conditional execution and looping
 - **Template** : which includes tags for producing and using common JSP templates in multiple pages

Custom Tags Benefits

- Using these custom tags, fields are automatically populated from the HTML form into the Form Bean
- JSPs only needs to know the proper field names and where to submit the form
- HTML input field code such as:

```
<input type="text" name="amount" value="<%=beanName.getAmount() %>">
```

- Can be written as:

```
<html:text property="amount"/>
```

Struts JSP html Tags

- Useful Presentation Tags
 - [html] [link](#) generates a HTML <a> element as an anchor definition or a hyperlink to the specified URL, and automatically applies URL encoding to maintain session state in the absence of cookie support
 - [html] [img](#) generates a HTML element with the ability to dynamically modify the URLs
 - [bean] [parameter](#) retrieves the value of the specified request parameter

HTML input form tags

- button
- cancel
- checkboxes
- file
- hidden
- image
- multibox
- password input fields
- radio buttons
- reset buttons
- select lists with embedded
- option
- options
- submit buttons
- text input fields
- textareas

HTML Form

```
<HTML>
<BODY>
<FORM>
<TABLE WIDTH="100%">
  <TR><TD>First Name</TD>
    <TD><INPUT TYPE="TEXT" NAME="Name" SIZE="40"
MAXLENGTH="40"></TD></TR>
  <TR><TD>Street Address</TD>
    <TD><INPUT TYPE="TEXT" NAME="Address" SIZE="40" MAXLENGTH="40"></TD></TR>
  <TR><TD>City</TD>
    <TD><INPUT TYPE="TEXT" NAME="City" SIZE="20" MAXLENGTH="20"></TD></TR>
  <TR><TD>State</TD>
    <TD><INPUT TYPE="TEXT" NAME="State" SIZE="2" MAXLENGTH="2"></TD></TR>
  <TR><TD>Postal Code</TD>
    <TD><INPUT TYPE="TEXT" NAME="ZipCode" SIZE="9" MAXLENGTH="9"></TD></TR>
  <TR><TD ALIGN="CENTER"><INPUT TYPE="SUBMIT" NAME="Submit" VALUE="Save"></TD>
    <TD><INPUT TYPE="RESET" NAME="Reset" VALUE="Cancel"></TD></TR>
</TABLE>
</FORM>
</BODY>
</HTML>
```

Struts HTML Form

```
<HTML:HTML>
<BODY>
<HTML:FORM Action="/RegistrationAction.do" focus="name" >
<TABLE WIDTH="100%">
  <TR><TD><bean:message key="customer.name"/></TD>
    <TD><HTML:TEXT property="name" size="40"
maxlength="40" /></TD></TR>
  <TR><TD><bean:message key="customer.address"/></TD>
    <TD><HTML:TEXT property="address" size="40" maxlength="40" /></TD></TR>
  <TR><TD><bean:message key="customer.city"/></TD>
    <TD><HTML:TEXT property="city" size="20" maxlength="20" /></TD></TR>
  <TR><TD><bean:message key="customer.state"/></TD>
    <TD><HTML:TEXT property="state" size="2" maxlength="2" /></TD></TR>
  <TR><TD><bean:message key="customer.zip"/></TD>
    <TD><HTML:TEXT property="zip" size="9" maxlength="9" /></TD></TR>
  <TR><TD ALIGN="CENTER"><html:submit property="action" value="Save"/></TD>
    <TD><html:reset property="action" value="Reset"/></TD></TR>
</TABLE>
</HTML:FORM>
</BODY>
</HTML:HTML>
```

Struts JSP bean Tags

- Struts bean tags can provide the following:
 - Create new bean from response to another request, from an application resource, or from a Struts configuration object
 - Write a localized message for the current user from a properties file
 - Write the value of a given property on any available bean
 - Create a JSP scripting variable from an HTTP header, request parameter, cookie, or any existing object
- Common bean tags are the `<bean:write>` and `<bean:message>`

Struts JSP bean Tags

```
<table border="2">
<tr>
  <th align="left"><b>bean:message key="hello.world"/></th>
</tr>
<logic:iterate id="image" property="collection"
  name="ImageLocationListForm">
<tr>
<td><a href="ImageLocationListForm.do?lob=<b>bean:write name='image'
  property='lob'/>
  &unitnbr=<bean:write name='image' property='unitnbr'/>
  &onbase_dns=<bean:write name='image' property='onbase_dns'/>" >
  <bean:write name="image" property="lob"/> </a></td>
</tr>
</logic:iterate>
</table>
```

Struts JSP logic Tags

- The Logic tag library contains tags that are useful in managing conditional generation of output text, looping over object collections for repetitive generation of output text, and application flow management.

Struts JSP logic Tags

- Useful Struts JSP logic Tags
 - [logic] iterate repeats its tag body once for each element of a specified collection (which can be an Enumeration, a Hashtable, a Vector, or an array of objects)
 - [logic] present depending on which attribute is specified, this tag checks the current request, and evaluates the nested body content of this tag only if the specified value is present
 - [logic] notPresent the companion tag to present, notPresent provides the same functionality when the specified attribute is not present

Struts JSP logic Tags

```
<html:html>
<head>
<title><bean:message key="imagebrokerlink.title"/></title>
<META name="GENERATOR" content="IBM WebSphere Studio">
</head>
<body>
<html:form action="/RegistrationAction.do" >
<center>
<font size=3>
<br>
<b>
<logic:notEqual property="action" name="ImageLocationForm"
value="Insert">
<bean:message key="imagelocationdetail.title"/>
</logic:notEqual>
<logic:equal property="action" name="ImageLocationForm"
value="Insert">
<bean:message key="imagelocationinsert.title"/>
</logic:equal>
...
...

```

Struts JSP template Tags

- Leveraging layouts
 - `<jsp:include>`
 - Struts JSP template Tags
 - Using Tiles with Struts (Tiles included in Struts 1.1)
- Master template provides a layout for a page

Struts JSP template Tags

- **Insert** Inserts (includes, actually) a template. Templates are JSP pages that include parameterized content. That content comes from put tags that are children of insert tags.
- **Put** Puts content into request scope.
- **Get** Gets the content from request scope that was put there by a put tag.

HTML Forms with Struts

- Review: Building Struts HTML Forms
 - The `taglib` directive tells the JSP page compiler where to find the *tag library descriptor* for the Struts tag library
 - `message` tag is used to look up internationalized message strings from a `MessageResources` object containing all the resources for this application
 - The `errors` tag displays any error messages that have been stored by a business logic component, or nothing if no errors have been stored

HTML Forms with Struts

- Review: Building Struts HTML Forms
 - The form tag renders an HTML `<form>` element, based on the specified attributes
 - The form bean can be specified in the Struts configuration file
 - The text tag renders an HTML `<input>` element of type "text"
 - The submit and reset tags generate the corresponding buttons for the form

Example Struts JSP

```
<%@ page language="java" %>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean" %>
<html:html>
<head> <title> <bean:message key="logon.title"/> </title>
<body bgcolor="white">
  <html:errors/>
<html:form action="/logonpath.do">
<table border="0" width="100%">
<tr> <th align="right"> <bean:message key="prompt.username"/>
  </th>
<td align="left"> <html:text property="username" size="16"/>
  </td> </tr>
<tr> <td align="right"> <html:submit> <bean:message
  key="button.submit"/> </html:submit> </td>
```

- Automatic Form Validation
 - Struts offers an additional facility to validate the input fields it has received
 - To utilize this feature, override the `validate()` method in your `ActionForm` class
 - The `validate()` method is called by the controller servlet after the bean properties have been populated, but before the corresponding action class's `perform()` method is invoked

Struts Configuration File

- /WEB-INF/struts-config.xml
- Configuration file contains action mappings (determines navigation)
- Controller uses mappings to call Struts Actions depending on the HTTP requests
- Mapping must specify
 - A request path
 - Object type to act upon the request

Struts Configuration File

```
<?xml version="1.0" encoding="ISO-8859-1" ?>  
<!DOCTYPE struts-config PUBLIC  
"-//Apache Software Foundation//DTD Struts  
Configuration 1.0//EN"  
"http://jakarta.apache.org/struts/dtds/struts-  
config_1_0.dtd">  
<struts-config>  
<!-- Here come the application elements  
declarations -->  
</struts-config>
```

Struts Form Bean Mappings

- **<form-beans>**

This section contains your form bean definitions. You use a `<form-bean>` element for each form bean, which has the following important attributes:

- **name**: The name of the request or session level attribute that this form bean will be stored as
- **type**: The fully-qualified Java classname of your form bean

Struts Action Mappings

- **<action-mappings>**

This section contains your action definitions. You use an `<action>` element for each of your actions you would like to define. Each action element has requires the following attributes to be defined:

- **path**: The application context-relative path to the action
- **type**: The fully qualified java classname of your Action class
- **name**: The name of your `<form-bean>` element to use with this action

Struts Mappings

In JSP /login.jsp:

```
<html:form action="/login.do" focus="user">  
  <html:text property="user" size="20" maxlength="20"/>  
  <html:password property="password" size="20" maxlength="20"  
  redisplay="false"/>  
  <html:submit property="submit" value="Login"/>  
  <html:reset/>  
</html:form>
```

The above code will have a corresponding mapping in struts-config.xml:

```
<struts-config>  
  <form-beans>  
    <form-bean name="loginForm"  
      type="com.capstonec.examples.LoginForm"/>  
  </form-beans>  
  <action-mappings>  
    <action path="/login" type="com.capstonec.exmaples.LoginAction"  
      name="loginForm" input="/login.jsp">  
      <forward name="success" redirect="true" path="/menu.jsp" />  
      <forward name="error" redirect="true" path="/error.jsp" />  
    </action>  
  </action-mappings>  
</struts-config>
```

Application.properties

error.na=[n/a]
error.int=[Invalid number]
error.null=[required]
error.day=[dd]
error.month=[mm]
error.year=[yyyy]
error.date=[invalid date (dd-mm-yyyy)]
error.minTwoletter=[min 2 letters]
error.email=[invalid email]
error.double=[invalid number]
error.text=[You are limited to 50 characters]

Internationalization

- Internationalized Messages
 - Struts builds upon Java platform to provide assistance for building internationalized and localized applications
 - Locale - fundamental Java class that supports internationalization
 - ResourceBundle - supports messages in multiple languages
 - PropertyResourceBundle - standard implementation of ResourceBundle that allows you to define resources using the same "name=value" syntax used to initialize properties files
 - MessageFormat - allows you to replace portions of a message string with arguments specified at run time
 - MessageResources - lets you treat a set of resource bundles like a database, and allows you to request a particular message string for a particular Locale

Internationalization

- `ApplicationResources.properties`
 - Contains the messages in the default language for your server. If your default language is English, you might have an entry like this: `prompt.hello=Hello`
- `ApplicationResources_xx.properties`
 - Contains the same messages in the language whose ISO language code is "xx" (i.e. `ApplicationResources_es.properties` for Spanish)

Building Struts Applications

- Your primary responsibilities are:
 - Update the web application deployment descriptor file (in XML) for your application to include the necessary Struts components
 - Add the appropriate Struts components to your application (taglibs)
 - Write the Struts Action class (that is, an extension of the Action class) for each logical request that may be received
 - Write the FormBeans (setters and getters)
 - Write the configuration file (in XML) that is used to configure the controller servlet (struts-config.xml)

- LookupDispatchAction
 - Encode several business-related methods within a single Action class (add, update, & delete)
 - *http://myhost/ShoppingCartAction?method=addItem*
- DynaActionForm
 - Declare a form bean completely through the struts configuration file
 - Eliminate the need to code form beans with setters and getters
- nested taglib
 - The ability to render values from a collection of collection of objects

Struts 1.1 : DynaActionForm

The following example creates a dynamic ActionForm called logonForm that contains two instance variables: username and password

```
<form-bean name="logonForm"  
  type="org.apache.struts.action.DynaAction  
  Form">  
  <form-property name="username"  
    type="java.lang.String" />  
  <form-property name="password"  
    type="java.lang.String" />  
</form-bean>
```

Struts 1.1 : Tiles

- Tiles now included with Struts 1.1
- Advanced templating features
- Tiles framework refers to templates as tiles
- Configuration of template definitions in a file (name of file can be specified in the web.xml)
- Using definitions as ActionForwards

Struts GUI Tools

Name	Cost	URL
Adalon	Commerical	http://www.synthis.com/products/adalon
Easy Struts	Open Source	http://easystruts.sourceforge.net/
Struts Console	Free	http://www.jamesholmes.com/struts/console
JForms	Commerical	http://www.solanasoft.com/
Camino	Commerical	http://www.scioworks.com/scioworks_camino.html
Struts Builder	Open Source	http://sourceforge.net/projects/rivernorth/
StrutsGUI	Free	http://www.alien-factory.co.uk/struts/struts-index.html

Struts Books

Name	Author
Programming Jakarta Struts	Chuck Canvaness
Struts In Action	Ted Husted
Struts Kick Start	James Turner, Kevin Bedell
Mastering Jakarta Struts	James Goodwill

Questions & Answers



CAPSTONE
Building **Solid** Enterprise **Solutions**

Download the slides at:
<http://www.capstoneC.com/>