

# Struts

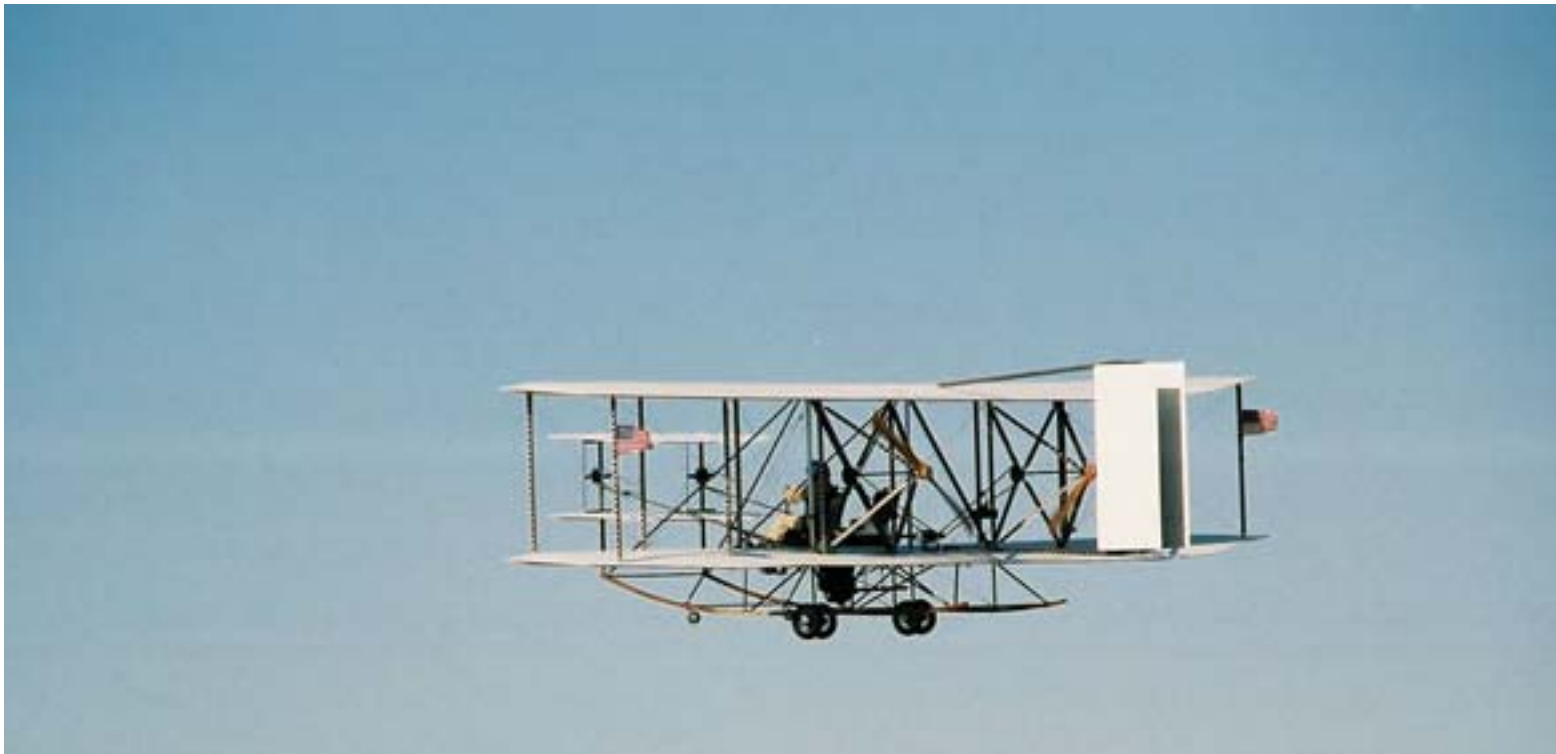
What they are, and How we use 'em

---

PEG Lunch-n-Learn

Nov 9, 2005

# Struts are support mechanisms





# What is Struts?

---

- MVC-based (Model-View-Controller) open source software
- Hosted by Apache Software Foundation
- Used to support the building of business applications

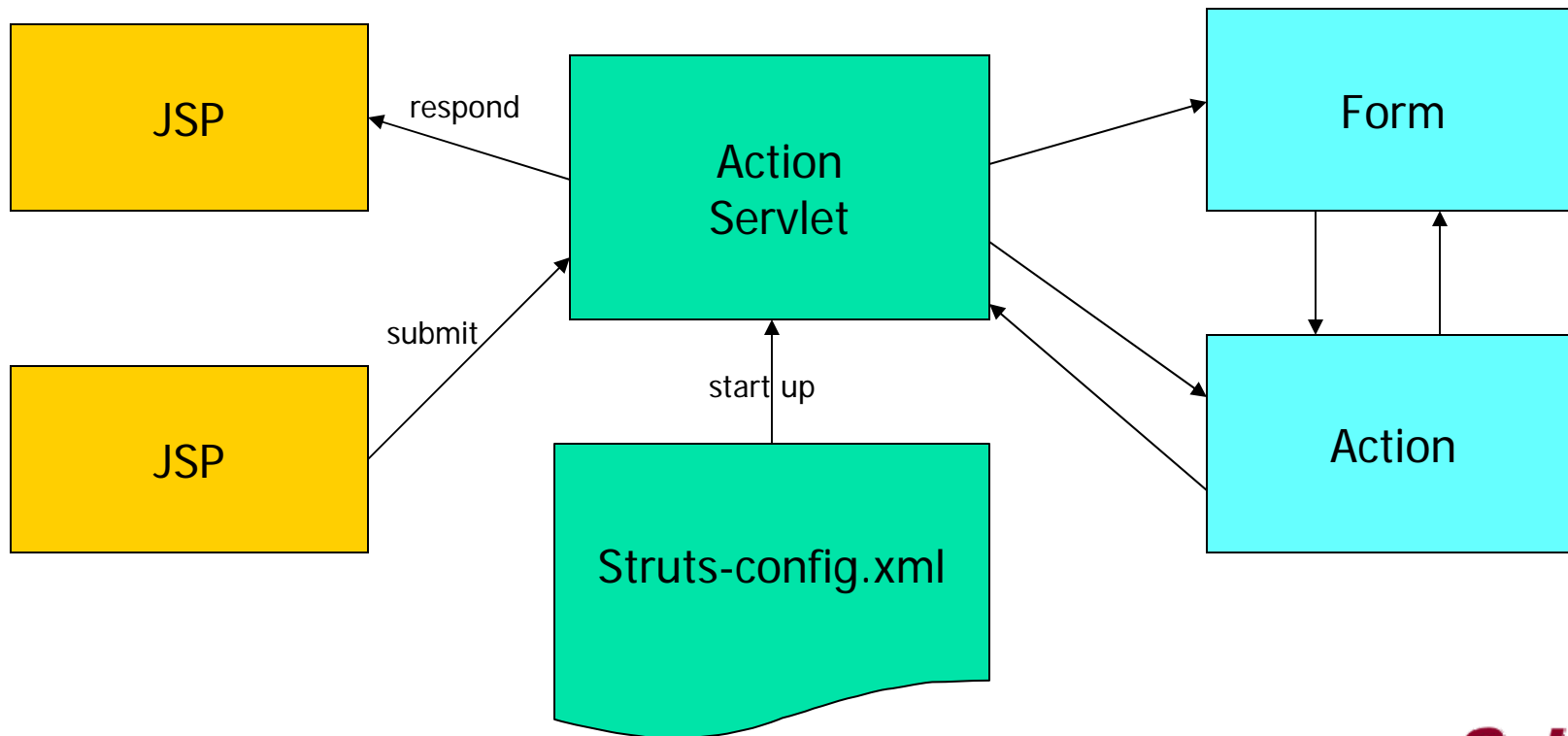


# The Big Picture

---

- **Struts-config.xml** - provides configuration information to the Action Servlet for all Struts objects that will be used by the application.
- **Action Servlet** - controls navigational flow.
- **Action Class** - accesses the business classes.
- **Action Form** - validates form inputs; provides access for the action class to the request parameters; used by action class to store and retrieve data.
- **ActionForward** - forwards the request to the specified path.

# The Big Picture





# Application Set-up

---

- Jar files
- Tag libraries
- Struts-config.dtd
- Application.properties
- Apache vhost file
- Web.xml
- Struts-config.xml



# Application Set-up (cont.)

---

- Jar files
  - Place in \WEB-INF\lib
- Tag Libraries
  - Place in \WEB-INF>taglibs
- Struts-config dtd
  - Place in \WEB-INF



# Application Set-up (cont.)

---

- Application.properties
  - Default location: \WEB-INF\classes\resources
  - Somewhat configurable
    - Name can be "name".properties
    - Location can be \WEB-INF\classes\"folder name"
    - Currently using "bundles" folder name
  - Contains messages for display within the application



# Application Set-up (cont.)

---

- Apache vhost file:

```
RewriteRule ^/TicketManagement/?$ /TicketManagement/soComLogin.jsp [R]
```

```
<Directory "C:/sites/engdev/tomcat/webapps/TicketManagement">  
  DirectoryIndex soComLogin.jsp  
  Order deny,allow  
  Allow from all  
</Directory>
```

### These JkMount directives will determine what requests will be sent  
### to Tomcat.

```
JkMount /TicketManagement/*.do ajp13
```

```
JkMount /TicketManagement/*.jsp ajp13
```



# Application Set-up (cont.)

---

- Web.xml file
  - Action servlet configuration
  - Servlet request mapping
  - Struts tag libraries configuration



# Web.xml Set-up

---

## ■ Servlet configuration:

```
<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/conf/struts-config.xml</param-value>
  </init-param>
  <init-param>
    <param-name>debug</param-name>
    <param-value>2</param-value>
  </init-param>
  <init-param>
    <param-name>detail</param-name>
    <param-value>2</param-value>
  </init-param>
  <load-on-startup>2</load-on-startup>
</servlet>
```



# Web.xml Set-up (cont.)

---

- Servlet request mapping:

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



# Web.xml Set-up (cont.)

---

- Struts tag libraries configuration:

```
<taglib>
    <taglib-uri>/taglibs/struts-bean</taglib-uri>
    <taglib-location>taglibs/struts-bean.tld</taglib-location>
</taglib>
<taglib>
    <taglib-uri>/taglibs/struts-html</taglib-uri>
    <taglib-location>taglibs/struts-html.tld</taglib-location>
</taglib>
<taglib>
    <taglib-uri>/taglibs/struts-logic</taglib-uri>
    <taglib-location>taglibs/struts-logic.tld</taglib-location>
</taglib>
<taglib>
    <taglib-uri>/taglibs/struts-nested</taglib-uri>
    <taglib-location>taglibs/struts-nested.tld</taglib-location>
</taglib>
```



# Application Set-up (cont.)

---

- Struts-config.xml
  - Location: \WEB-INF\conf
  - Basic configuration tags
    - <global-exceptions>
    - <form-beans>
    - <global-forwards>
    - <action-mappings>
    - <message-resources>
    - <plug-in>



# Struts-config.xml

---

- Global-exceptions
  - Allows exceptions to be handled in a consistent way by all Action objects.
  - Define exception handler to handle exceptions thrown by an Action object.
  - Format:

```
<global-exceptions>  
  <exception handler="some.package.InputExceptionHandler"  
    type="java.lang.Exception"  
    key=""  
    path="/exceptionDisplayPage.jsp" />  
</global-exceptions>
```



# Struts-config.xml (cont.)

---

- Form-beans

- Defines what ActionForm objects can be created by the ActionServlet, and what to call them.

- Format:

```
<form-beans>
  <form-bean name="inactivateContactForm"
    type="some.package.name.InactivateContactForm" />
  <form-bean name="UserSelfRegistrationActionForm"
    type="org.apache.struts.action.DynaActionForm">
    <form-property name="userName" type="java.lang.String" />
  </form-bean>
</form-beans>
```





# Struts-config.xml (cont.)

---

- Global-forwards
  - Defines ActionForward paths that are available to all Actions in an application.
  - Format:

```
<global-forwards>  
  <forward name="mergeSuccess" path="/MergeDisplay.do" />  
  <forward name="cancel" path="/soCacCloseTargetPage.jsp" />  
</global-forwards>
```



# Struts-config.xml (cont.)

---

- Action-mappings

- Define what operations the application can undertake.

```
<action-mappings>
  <action path="/SelectMergeJournal"
    type="some.package.name.SelectMergeJournalAction"
    name="selectMergeJournalForm"
    scope="request"
    validate="true"
    input="/soCacMergeJournalSelectSources.jsp" >
    <forward name="back" path="/StartMergeJournal.do" />
    <forward name="next" path="/FinishMergeJournal.do" />
  </action>
  <action path="/FinishMergeJournalDisplay"
    type="org.apache.struts.actions.ForwardAction"
    parameter="/soCacMergeJournalFinish.jsp" />
</action-mappings>
```



# Struts-config.xml (cont.)

---

- Message-resources
  - Deploys the message bundle files needed by the application (xyz.properties).
  - Default file:
    - \WEB-INF\classes\resources\application.properties
  - Format:

```
<message-resources key="spatial"  
    parameter="bundles.SpatialObjectsResourceBundle" />  
<message-resources key="cac"  
    parameter="bundles.CacResourceBundle" />
```



# Struts-config.xml (cont.)

---

- Plug-in

- Defines special resources that are available to the Action classes.
- Format:

```
<plug-in className="org.apache.struts.tiles.TilesPlugin" >  
  <set-property property="definitions-config" value="/WEB-INF/conf/tiles-def.xml" />  
  <set-property property="definitions-parser-validate" value="true" />  
  <set-property property="definitions-debug" value="2" />  
  <set-property property="definitions-parser-details" value="2" />  
</plug-in>
```



## Application Set-up (cont.)

---

- Set-up process is rather stringent.
- Struts expects certain files to be in specific folders in specific paths.



# Simple Example Request

---

- Request is sent to the ActionServlet
  - /Request.do?param1=value&param2=...
- Servlet checks the ActionMapping for Request.do
- Servlet instantiates the ActionForm
  - Form is populated with request parameters via setter methods, i.e. `setParam1(String pValue)`
  - Form performs any validation on the supplied values



## Simple Example Request (cont.)

---

- Servlet executes the specified Action class, passing it the ActionForm
- Action class does its duty, getting and/or setting data in the ActionForm
- Action forwards to the ActionServlet
- Servlet executes a JSP
- JSP displays data from the ActionForm

A Brief Look at –

# Struts HTML Tags

- A wide range of tags are available
  - Complete list and description found at:  
<http://struts.apache.org/struts-taglib/index.html> or  
<http://struts.apache.org/struts-doc-1.2.7/userGuide/index.html>
- Use can replace Java code in a JSP
- Can be somewhat “clunky” to use
- Not always applicable to coding needs

A Brief Look at –

## Struts HTML Tags (cont.)

- Example of form and message tags

```
<html:form method="post" action="/SelectMergeJournal">
  <table width="100%" cellspacing="0" cellpadding="0" border="0">
    <logic:messagesPresent>
      <tr>
        <td><html:errors bundle="cacMessages"/></td>
      </tr>
    </logic:messagesPresent>
    ...
  </table>
</html:form>
```

A Brief Look at –

## Struts HTML Tags (cont.)

- Example of submit and button tags

```
<html:submit value="Next" property="btnNext" style="width: 80px" />
```

```
<html:submit value="Back" property="btnBack" style="width: 80px" />
```

```
<html:button value="Cancel" property="btnCancel" style="width: 80px"  
onclick="<%= sCANCEL_PAGE_TARGET_ON_CLICK_SCRIPT %>" />
```

## A Brief Look at –

# Struts HTML Tags (cont.)

- Example of logic and select-box tags

```
<logic:equal value="true" name="manageTicketForm" property="hasTicketHistory">
  <td class="soFieldLabel" style="vertical-align: middle;">Ticket History:</td>
  <td class="FieldLabelLeftAlign">
    <html:select name="manageTicketForm" property="selectedTicketId"
      styleId="selTicketHistory"
      onChange="selectTicketHistory('selTicketHistory');"
      value="<%= selectedTicketId %>" >
      <html:options name="manageTicketForm" property="ticketHistoryId"
        labelName="manageTicketForm"
        labelProperty="ticketHistoryLabel" />
    </html:select>
  </td>
</logic:equal>
```





# Thread Safety

---

- Servlet creates one instance of each Action class per application.
- Only use local variables, not instance (class) variables.
- Shared values must be passed through the method signatures.



# Modular Applications

---

- Multiple modules can run within the same application space
- Each module has its own prefix
- Prefix corresponds to the folder name where the module files reside
- Module configuration is in web.xml



# Modular Applications (cont.)

---

## ■ Web.xml setup

```
<init-param>
```

```
  <param-name>config</param-name>
```

```
  <param-value>/WEB-INF/conf/struts-config.xml</param-value>
```

```
</init-param>
```

```
<init-param>
```

```
  <param-name>config/adminModule</param-name>
```

```
  <param-value>/WEB-INF/conf/struts-adminModule.xml</param-value>
```

```
</init-param>
```

```
<init-param>
```

```
  <param-name>config/loginModule</param-name>
```

```
  <param-value>/WEB-INF/conf/struts-loginModule.xml</param-value>
```

```
</init-param>
```





# Modular Applications (cont.)

---

- Struts-loginModule.xml setup

```
<action-mappings>
  <action path="/ApplicationHome"
    type="org.apache.struts.actions.ForwardAction"
    parameter="/Switch.do?prefix=&page=/DataPortCart/soMainFrames.jsp" />
  <action path="/Login"
    type="org.apache.struts.actions.ForwardAction"
    parameter="/loginModule/soComLogin.jsp" />
  <action path="/Logout"
    type="org.apache.struts.actions.ForwardAction"
    parameter="/loginModule/soComLogout.jsp" />
  <action path="/Switch"
    type="org.apache.struts.actions.SwitchAction" />
</action-mappings>
```



# Modular Applications (cont.)

---

- Modules are referenced by "moduleName"/"action"
- Example of vhost file:

### do a redirect to the entry point for this application

RewriteEngine On

RewriteRule ^/TicketManagement/?\$ /TicketManagement/**loginModule/Login.do** [R]

### RewriteRule ^/TicketManagement/?\$ /TicketManagement/soComLogin.jsp [R]



# Modular Applications (cont.)

---

- Pros

- Modules can be developed/modified independently
- Smaller, cleaner configuration files

- Cons

- Global items (forwards, exception handlers, plug-ins) must be defined in each module that needs access to them



# Modular Applications (cont.)

---

- Possible work-around #1

```
<param-name>config</param-name >  
<param-value>  
    /WEB-INF/conf/struts-config-common.xml,  
    /WEB-INF/conf/struts-config.xml  
</param-value>  
<param-name>config/module2</param-name >  
<param-value>  
    /WEB-INF/conf/struts-config-common.xml,  
    /WEB-INF/conf/struts-config-module2.xml  
</param-value>
```



# Modular Applications (cont.)

---

- Possible work-around #2

```
<param-name>config</param-name >  
<param-value>  
    /WEB-INF/conf/struts-config.xml,  
    /WEB-INF/conf/struts-config-module1.xml,  
    /WEB-INF/conf/struts-config-module2.xml  
</param-value>
```



# Struts Tiles





# What is Struts Tiles

---

- Tiles is a templating library
- Allows construction of pages in interchangeable parts (tiles)
- Facilitates the creation of “reusable” view components



# Page Construction

---

- A page can be broken into sections which are implemented as tiles
- Each tile may have its own Action and Form class or may take part in others

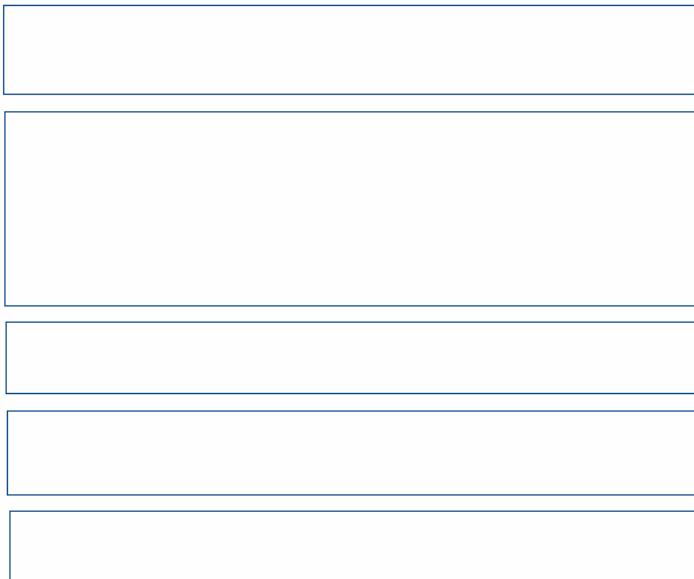




# A WebFast Example

---

- Webfast User Registration, User Administration and Security Administration all use tiles (in various combinations)





# Adding Tiles to the struts-config.xml

---

```
<plug-in className="org.apache.struts.tiles.TilesPlugin" >  
  <set-property property="definitions-config" value="/WEB-INF/conf/tiles-def.xml" />  
  <set-property property="definitions-parser-validate" value="true" />  
  <set-property property="definitions-debug" value="2" />  
  <set-property property="definitions-parser-details" value="2" />  
</plug-in>
```



# The Tile Configuration

---

```
<definition name="userMaintenanceDef"
  page="/cwWebFastUserAdministrationContainer.jsp">
  <put name="submitAction"
    value="/userMaintenance"
    type="string" />
  <put name="pageTitle"
    value="User Maintenance"
    type="string" />
  <put name="formName"
    value="UserMaintenanceActionForm"
    type="string" />
  <putList name="tiles">
    <add value="userMaintenanceIdentityDef" type="definition" />
    <add value="userSelfRegistrationContactDef" type="definition" />
    <add value="userMaintenanceGroupsDef" type="definition" />
    <add value="userMaintenanceRolesDef" type="definition" />
    <add value="userAdministrationStatusDef" type="definition" />
  </putList>
  <putList name="buttons">
    <add value="updateButtonDef" type="definition"/>
  </putList>
</definition>
```



# Grouting the Tiles

---

```
<form-bean name="UserMaintenanceActionForm"
  type="org.apache.struts.action.DynaActionForm">
  <form-property name="userName" type="java.lang.String" />
  <form-property name="identityActionForm"
    type="com.celeritasworks.webfast.sec.struts.forms.UserSelfRegistrationIdentityActionForm" />
  <form-property name="contactActionForm"
    type="com.celeritasworks.webfast.sec.struts.forms.UserSelfRegistrationContactActionForm" />
  <form-property name="groupsActionForm"
    type="com.celeritasworks.webfast.sec.struts.forms.UserMaintenanceGroupsActionForm" />
  <form-property name="rolesActionForm"
    type="com.celeritasworks.webfast.sec.struts.forms.UserMaintenanceRolesActionForm" />
  <form-property name="statusActionForm"
    type="com.celeritasworks.webfast.sec.struts.forms.UserAdministrationStatusActionForm" />
</form-bean>

<action path="/userMaintenance"
  type="com.celeritasworks.webfast.sec.struts.actions.UserMaintenanceAction"
  name="UserMaintenanceActionForm"
  scope="request"
  validate="false"
  input="userMaintenanceDef">
  <forward name="success" path="userMaintenanceDef" />
</action>
```



# Celeritas Technologies

---



7101 College Blvd, Sixth Floor  
Overland Park, KS 66210  
913.491.9000

[www.celeritas.com](http://www.celeritas.com)

<http://java.celeritas.com>