STRUTS
CROSS
REFERENCE

Prepared by
Roger W. Barnes
Of
Project Refinery, Inc.

TABLE OF CONTENTS

OVERVIEW **3**
STRUTS-CONFIG.XML **4**
WEB.XML **5**
STRUTSINPUTFORM.JAVA **6**
STRUTSINPUTACTION.JAVA **9**
PGINPUT.JSP **10**
PGDISPLAY.JSP **11**
STRUTSDISPLAYFORM.JAVA **12**
STRUTSDISPLAYACTION.JAVA **13**
STRUTSEXAMPLE.PROPERTIES FILE **14**
TYPICAL STRUTS SYSTEM FLOW **15**

Overview

The purpose of this document is to provide a cross reference of an example application using the Struts framework. This document displays where information from the struts-config.xml file relate to Java Server Pages using the Struts tag libraries, and to Java classes extending both the ActionForm and Action classes.

(11)

Struts-config.xml

```
<?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>

  <!-- ========== Form Bean Definitions ================================== -->
  <form-beans>

    <!-- Struts Example form bean -->

    <form-bean      name="StrutsInputFormBean"  (1)

                    type="org.struts.example.StrutsInputForm"  (4)  />


    <form-bean      name="StrutsDisplayFormBean"  (2)

                    type="org.struts.example.StrutsDisplayForm"  (8) />

  </form-beans>

  <!-- ========== Global Forward Definitions ============================== -->
  <global-forwards>
    <forward  name="Menu"                        path="/index.html"/>
  </global-forwards>
  <!-- ========== Action Mapping Definitions ============================== -->
  <action-mappings>

    <!-- StrutsExample -->

    <action    path="/StrutsInputPath"  (3)

               type="org.struts.example.StrutsInputAction"  (5)

               name="StrutsInputFormBean"  (1)
              scope="request"

              input="/PgInput.jsp">  (6)

      <forward name="Next"               path="/PgDisplay.jsp"  (7) />
    </action>

    <action    path="/StrutsDisplayPath"  (9)


               type="org.struts.example.StrutsDisplayAction"  (10)

               name="StrutsDisplayFormBean"  (2)
              scope="request"

              input="/PgDisplay.jsp"  (7) >
    </action>

  </action-mappings>
</struts-config>
```

web.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE web-app
  PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.2//EN"
  "http://java.sun.com/j2ee/dtds/web-app_2_2.dtd">
<web-app id="WebApp">
  <display-name>StrutsExampleWeb</display-name>
  <!-- Action Servlet Configuration -->
  <servlet id="Servlet_1">
    <servlet-name>action</servlet-name>
    <servlet-class>org.apache.struts.action.ActionServlet</servlet-class>
    <init-param>
      <param-name>application</param-name>

      <param-value>strutsexample (12) </param-value>
    </init-param>
    <init-param>
      <param-name>config</param-name>

      <param-value>WEB-INF/struts-config.xml (11) </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>
    <init-param>
      <param-name>validate</param-name>
      <param-value>true</param-value>
    </init-param>
    <load-on-startup>2</load-on-startup>
  </servlet>

  <!-- Action Servlet Mapping -->
  <servlet-mapping>
    <servlet-name>action</servlet-name>
    <url-pattern>*.do</url-pattern>
  </servlet-mapping>


  <!-- The Welcome File List -->
  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
  </welcome-file-list>


  <!-- Tag Library Descriptors -->
  <taglib>
    <taglib-uri>struts</taglib-uri>
    <taglib-location>/WEB-INF/lib/struts.jar</taglib-location>
  </taglib>
  <taglib>
    <taglib-uri>WEB-INF/struts-bean.tld</taglib-uri>
    <taglib-location>/WEB-INF/struts-bean.tld</taglib-location>
  </taglib>
  <taglib>
    <taglib-uri>WEB-INF/struts-html.tld</taglib-uri>
    <taglib-location>/WEB-INF/struts-html.tld</taglib-location>
  </taglib>
  <taglib>
    <taglib-uri>WEB-INF/struts-logic.tld</taglib-uri>
    <taglib-location>/WEB-INF/struts-logic.tld</taglib-location>
  </taglib>
  <taglib>
    <taglib-uri>WEB-INF/struts-form.tld</taglib-uri>
    <taglib-location>/WEB-INF/struts-form.tld</taglib-location>
  </taglib>
  <taglib>
    <taglib-uri>WEB-INF/struts-template.tld</taglib-uri>
    <taglib-location>/WEB-INF/struts-template.tld</taglib-location>
  </taglib>
</web-app>
```

StrutsInputForm.java

```
package org.struts.example;

import javax.servlet.http.HttpServletRequest;
import org.apache.struts.action.ActionError;
import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionMapping;

/**
 * The purpose of this class is to handle the input fields entered on a JSP.
 * @author Roger W Barnes
 * @version $Revision: 1.0 $ $Date: 2001/11/02 19:55:25 $
 */
public final class StrutsInputForm extends ActionForm {
    /**
     * The name.
     */
    private String name = "";
    /**
     * The address.
     */
    private String address = "";
    /**
     * The city.
     */
    private String city = "";
    /**
     * The state.
     */
    private String state = "";
    /**
     * The zipcode.
     */
    private String zipcode = "";
    /**
     * The action.
     */
    private String action = "";

       /** END OF VARIABLES */

       /**
        * Gets the name
        * @return Returns a String
        */
       public String getName() {
              return name;
       }
       /**
        * Sets the name
        * @param name The name to set
        */
       public void setName(String name) {
              this.name = name;
       }
       /**
        * Gets the address
        * @return Returns a String
```

```
 */
public String getAddress() {
	return address;
}
/**
 * Sets the address
 * @param address The address to set
 */
public void setAddress(String address) {
	this.address = address;
}
/**
 * Gets the city
 * @return Returns a String
 */
public String getCity() {
	return city;
}
/**
 * Sets the city
 * @param city The city to set
 */
public void setCity(String city) {
	this.city = city;
}
/**
 * Gets the state
 * @return Returns a String
 */
public String getState() {
	return state;
}
/**
 * Sets the state
 * @param state The state to set
 */
public void setState(String state) {
	this.state = state;
}
/**
 * Gets the zipcode
 * @return Returns a String
 */
public String getZipcode() {
	return zipcode;
}
/**
 * Sets the zipcode
 * @param zipcode The zipcode to set
 */
public void setZipcode(String zipcode) {
	this.zipcode = zipcode;
}



/**
 * Gets the action
 * @return Returns a String
 */
public String getAction() {
	return action;
```

```
        }
        /**
         * Sets the action
         * @param action The action to set
         */
        public void setAction(String action) {
                this.action = action;
        }
    /**
     * Reset all properties to their default values.
     *
     * @param mapping The mapping used to select this instance
     * @param request The servlet request we are processing
     */
    public void reset(ActionMapping mapping, HttpServletRequest request) {
        action              = null;
    }
    /**
     * Validate the properties of this form bean, and return an array of
     * message keys for any errors we encounter.
     */
    public ActionErrors validate(ActionMapping mapping,
                                 HttpServletRequest request) {
        ActionErrors errors = new ActionErrors();
        /** Perform validation tests on the name field */
        if ((name == null) || (name.length() < 1))
            errors.add("name", new ActionError("error.name.required", name));
        /** Perform validation tests on the address field */
        if ((address == null) || (address.length() < 1))
            errors.add("address", new ActionError("error.address.required", address));
        /** Perform validation tests on the city field */
        if ((city == null) || (city.length() < 1))
            errors.add("city", new ActionError("error.city.required", city));
        /** Perform validation tests on the state field */
        if ((state == null) || (state.length() < 1))
            errors.add("state", new ActionError("error.state.required", state));
        /** Perform validation tests on the zipcode field */
        if ((zipcode == null) || (zipcode.length() < 1))
            errors.add("zipcode", new ActionError("error.zipcode.required", zipcode));
        /** Perform validation tests on the action field */
        if ((action == null) || (action.length() < 1))
            errors.add("action", new ActionError("error.action.required", action));
        return errors;
    }
}
```

⑤

StrutsInputAction.java

package org.struts.example;

```
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;

/**
 * Implementation of <strong>Action</strong> that handles the Index Account Action.
 *
 * @author Roger W Barnes
 * @version $Revision: 1.0 $ $Date: 2002/03/11 16:03:25 $
 */

public final class StrutsInputAction extends Action {
    /**
     * Process the specified HTTP request, and create the corresponding HTTP
     * response (or forward to another web component that will create it).
     * Return an <code>ActionForward</code> instance describing where and how
     * control should be forwarded, or <code>null</code> if the response has
     * already been completed.
     *
     * @param servlet The ActionServlet making this request
     * @param mapping The ActionMapping used to select this instance
     * @param actionForm The optional ActionForm bean for this request (if any)
     * @param request The HTTP request we are processing
     * @param response The HTTP response we are creating
     *
     * @exception IOException if an input/output error occurs
     * @exception ServletException if a servlet exception occurs
     */
    public ActionForward perform(ActionMapping mapping,
                                 ActionForm form,
                                 HttpServletRequest request,
                                 HttpServletResponse response)
        {
                    /** Get a handle to the StrutsInputForm */
                    StrutsInputForm strutsForm = (StrutsInputForm) form;

                    /** Get the action from the form */
            String action = strutsForm.getAction();

            /** Forward control to the specified success URI */
            return (mapping.findForward(action));

        }
}
```

⑥

PgInput.jsp

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<%@ page language="java" %>
<%@ taglib uri="WEB-INF/struts-bean.tld" prefix="bean" %>
<%@ taglib uri="WEB-INF/struts-html.tld" prefix="html" %>
<html:html>
<head>
<title><bean:message key="strutsexample.title1"/></title>
<META name="GENERATOR" content="IBM WebSphere Studio">
</head>
<body>

<html:form action="/StrutsInputPath" ③ focus="name" >
<table width=750 cellspacing=0 cellpadding=2 border=2 >
<tr><td><font color="blue"><bean:message key="strutsexample.name"/>:</font>
    </td>
       <td>   <html:text property="name" size="40" maxlength="40"/>  </td>
</tr>
<tr>   <td> </td>
       <td><font color=red><html:errors property="name"/></font></td>
</tr>
<tr>   <td><font color="blue"><bean:message key="strutsexample.address"/>:
              </font></td>
       <td><html:text property="address" size="40" maxlength="40"/></td>
</tr>
<tr>   <td> </td>
       <td><font color=red><html:errors property="address"/>  </font></td>
</tr>
<tr>   <td><font color="blue"><bean:message key="strutsexample.city"/>:</font>
       </td>
       <td>   <html:text property="city" size="20" maxlength="20"/>  </td>
</tr>
<tr>   <td> </td>
       <td><font color=red><html:errors property="city"/></font></td>
</tr>
<tr>   <td><font color="blue"><bean:message key="strutsexample.state"/>:</font>
       </td>
       <td><html:text property="state" size="2" maxlength="2"/></td>
</tr>
<tr>   <td> </td>
       <td><font color=red><html:errors property="state"/></font></td>
</tr>
<tr>   <td><font color="blue"><bean:message key="strutsexample.zipcode"/>:
              </font></td>
       <td><html:text property="zipcode" size="10" maxlength="10"/></td>
</tr>
<tr>   <td> </td>
       <td><font color=red><html:errors property="zipcode"/>  </font></td>
</tr>
</table>

<table width=750 cellspacing=0 cellpadding=2 border=2 >
<tr>
       <td align="center"><html:submit property="action" value="Next" /></td>
       <td align="center"><html:submit property="action" value="Menu" /></td>
</tr>
</table>

</html:form>

</body>
</html:html>
```

⑦

PgDisplay.jsp

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<%@ page language="java" %>
<%@ taglib uri="WEB-INF/struts-bean.tld" prefix="bean" %>
<%@ taglib uri="WEB-INF/struts-html.tld" prefix="html" %>

<html:html>
<head>
<title><bean:message key="strutsexample.title2"/></title>
<META name="GENERATOR" content="IBM WebSphere Studio">
</head>

<body>


<html:form action="/StrutsDisplayPath" ⑨ focus="action" >

<table width=750 cellspacing=0 cellpadding=2 border=2 >
<tr>
       <td><font color="blue"><bean:message key="strutsexample.name"/>:</font>
       </td>
       <td>   <bean:write property="name" name="StrutsInputFormBean" />
       </td>
</tr>
<tr>
       <td><font color="blue"><bean:message key="strutsexample.address"/>:
       </font>
       </td>
       <td>   <bean:write property="address" name="StrutsInputFormBean" />
       </td>
</tr>
<tr>
       <td><font color="blue"><bean:message key="strutsexample.city"/>:</font>
       </td>
       <td>   <bean:write property="city" name="StrutsInputFormBean" />
       </td>
</tr>
<tr>
       <td><font color="blue"><bean:message key="strutsexample.state"/>:</font>
       </td>
       <td>   <bean:write property="state" name="StrutsInputFormBean" />
       </td>
</tr>
<tr>
       <td><font color="blue"><bean:message key="strutsexample.zipcode"/>:</font>
       </td>
       <td>   <bean:write property="zipcode" name="StrutsInputFormBean" />
       </td>
</tr>
</table>


<table width=750 cellspacing=0 cellpadding=2 border=2 >
<tr>
       <td align="center">
              <html:submit property="action" value="Menu" />
       </td>
</tr>
</table>

</html:form>

</body>
</html:html>
```

StrutsDisplayForm.java

```
package org.struts.example;

import javax.servlet.http.HttpServletRequest;
import org.apache.struts.action.ActionError;
import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionMapping;

/**
 * The purpose of this class is to handle the input fields entered on a JSP form page.
 * @author Roger W Barnes
 * @version $Revision: 1.0 $ $Date: 2001/11/02 19:55:25 $
 */
public final class StrutsDisplayForm extends ActionForm {
    /**
     * The action.
     */
    private String action = "";
       /**
        * Gets the action
        * @return Returns a String
        */
       public String getAction() {
              return action;
       }
       /**
        * Sets the action
        * @param action The action to set
        */
       public void setAction(String action) {
              this.action = action;
       }
   /**
     * Reset all properties to their default values.
     *
     * @param mapping The mapping used to select this instance
     * @param request The servlet request we are processing
     */
    public void reset(ActionMapping mapping, HttpServletRequest request) {
        action              = null;
    }
    /**
     * Validate the properties of this form bean, and return an array of
     * message keys for any errors we encounter.
     */
    public ActionErrors validate(ActionMapping mapping,
                                 HttpServletRequest request) {
        ActionErrors errors = new ActionErrors();
       /** Perform validation tests on the action field */
        if ((action == null) || (action.length() < 1))
            errors.add("action", new ActionError("error.action.required", action));
       return errors;
    }
}
```

10

StrutsDisplayAction.java

```
package org.struts.example;

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;

/**
 * Implementation of <strong>Action</strong> that handles the Index Account Action.
 *
 * @author Roger W Barnes
 * @version $Revision: 1.0 $ $Date: 2002/03/11 16:03:25 $
 */

public final class StrutsDisplayAction extends Action {
    /**
     * Process the specified HTTP request, and create the corresponding HTTP
     * response (or forward to another web component that will create it).
     * Return an <code>ActionForward</code> instance describing where and how
     * control should be forwarded, or <code>null</code> if the response has
     * already been completed.
     *
     * @param servlet The ActionServlet making this request
     * @param mapping The ActionMapping used to select this instance
     * @param actionForm The optional ActionForm bean for this request (if any)
     * @param request The HTTP request we are processing
     * @param response The HTTP response we are creating
     *
     * @exception IOException if an input/output error occurs
     * @exception ServletException if a servlet exception occurs
     */
    public ActionForward perform(ActionMapping mapping,
                                 ActionForm form,
                                 HttpServletRequest request,
                                 HttpServletResponse response)
        {
                    /** Get a handle to the StrutsExampleForm */
                    StrutsDisplayForm strutsForm = (StrutsDisplayForm) form;

                    /** Get the action from the form */
            String action = strutsForm.getAction();

            /** Forward control to the specified success URI */
            return (mapping.findForward(action));

        }
}
```

12

strutsexample.properties File

```
strutsexample.title1=Struts Example Project - Input Form
strutsexample.title2=Struts Example Project - Display Page
strutsexample.name=Enter Your Name
strutsexample.address=Enter Your Address
strutsexample.city=Enter Your City
strutsexample.state=Enter Your State
strutsexample.zipcode=Enter Your Zip Code
error.name.required=<li>You must enter Your Name.</li>
error.address.required=<li>You must enter Your Address.</li>
error.city.required=<li>You must enter Your City.</li>
error.state.required=<li>You must enter Your State.</li>
error.zipcode.required=<li>You must enter Your Zip Code.</li>
error.action.required=<li>You must click on the Next button.</li>
errors.header=
errors.footer=
```

Typical Struts System Flow

- A request comes in from a Java Server Page into the ActionServlet
- The ActionServlet having already read the struts-config.xml file, knows which form bean relates to this JSP, and delegates work to the validate method of that form bean
- The form bean performs the validate method to determine if all required fields have been entered, and performs whatever other types of field validations that need to be performed
- If any required field has not been entered, or any field does not pass validation, the form bean generates ActionErrors, and after checking all fields returns back to the ActionServlet
- The ActionServlet checks the ActionErrors that were returned from the form bean's validate method to determine if any errors have occurred. If errors have occurred, it returns to the originating JSP displaying the appropriate errors
- If no errors occurred in the validate method of the form bean, the ActionServlet passes control to the appropriate Action class
- The Action class performs any necessary business logic, and then forwards to the next appropriate action (probably another JSP)