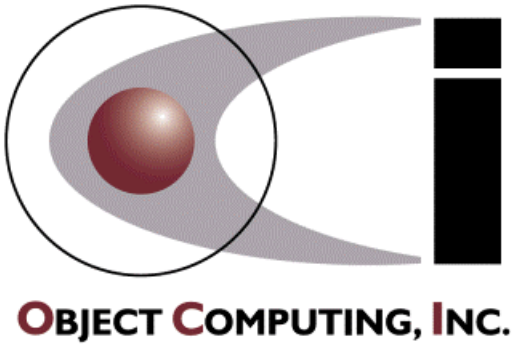


Jakarta Struts

Presented by
Object Computing, Inc. (OCI)
http://www.ociweb.com

Written by Greg Elliott
elliott_g@ociweb.com

What is Struts?

- An open source development framework for building web applications
- Based on Model-View-Controller (MVC) design paradigm
- Implementation of JSP Model 2 Architecture
- Created by Craig McClanahan and donated to Apache Software Foundation (ASF) in 2000
- 2nd release candidate of version 1.1 released
- Consists of 8 Top-Level Packages
- Approximately 250 Classes and Interfaces

Alternatives to Struts

- No framework (use straight JSP)
- Build your own framework
- Webwork
- Expresso
- Barracuda
- Cocoon
- SiteMesh
- Freemarker, Velocity and WebMacro
- XML/XSLT
- ???

Why consider Struts?

- Developed by industry experts
- Stable & Mature
- Manageable learning curve
- Open source
- Probably similar to what you would build if you weren't going to use Struts
- Good documentation – both javadoc api as well as numerous books on topic
- Feature-rich
- Supported by many 3rd party tools
- Flexible and extendable

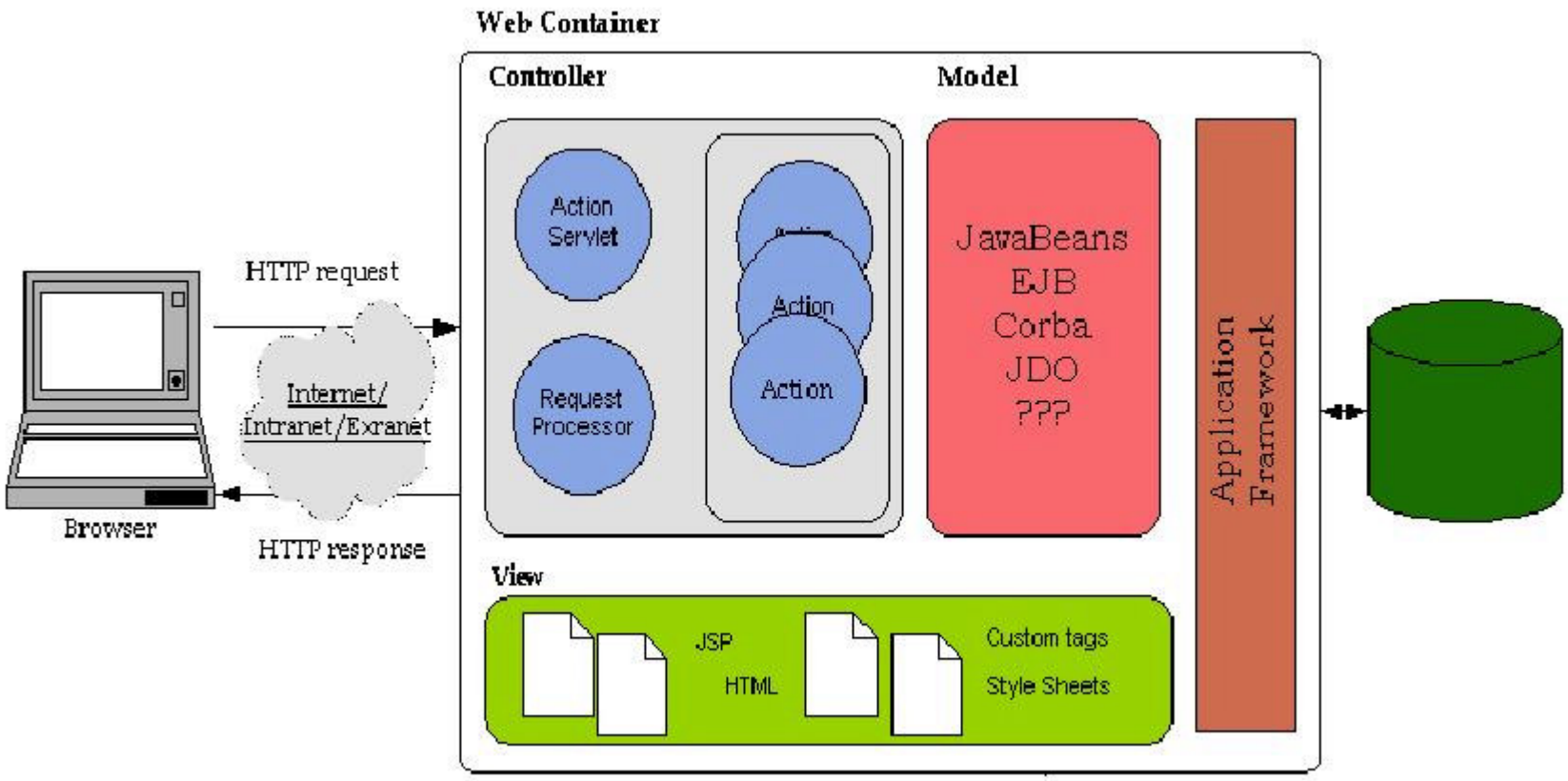
Struts Framework Features

- Model 2 – MVC Implementation
- Internationalization support
- Rich JSP tag libraries
- Based on JSP, Servlet, XML and Java
- Supports different model implementations (JavaBeans, EJB, OJB, etc.)
- Supports different presentation implementations (JSP, XML/XSLT, etc.)

Struts Dependencies

- Java 1.2 or newer
- Servlet 2.2 and JSP 1.1 container
- XML parser compliant with JAXP 1.1 or newer (ie, Xerces)
- Jakarta Commons packages
- JDBC 2.0 optional package

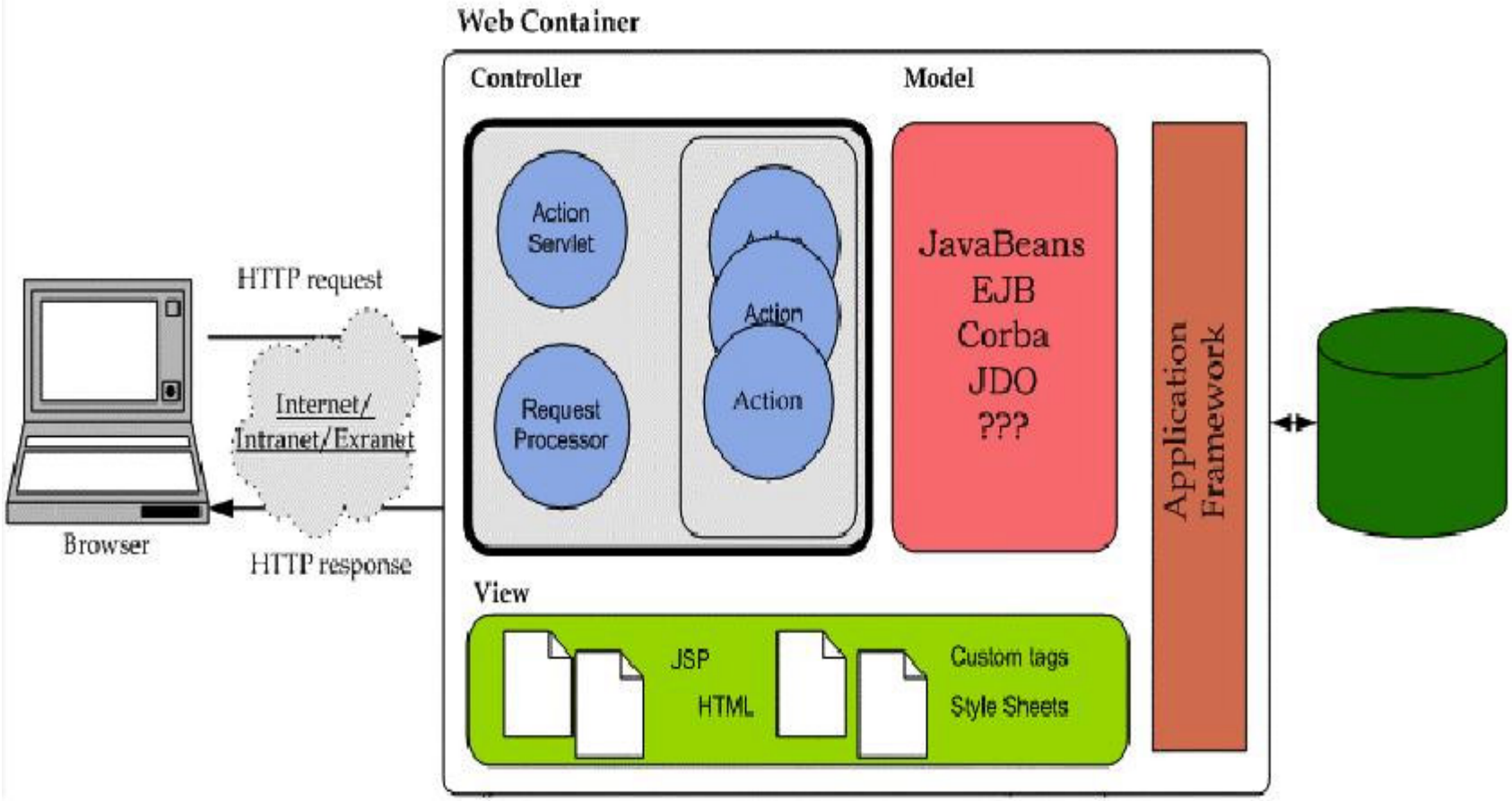
Logical Architecture



Aspects of the Framework

- Controller
- Model
- View
- Configuration issues

Controller Components



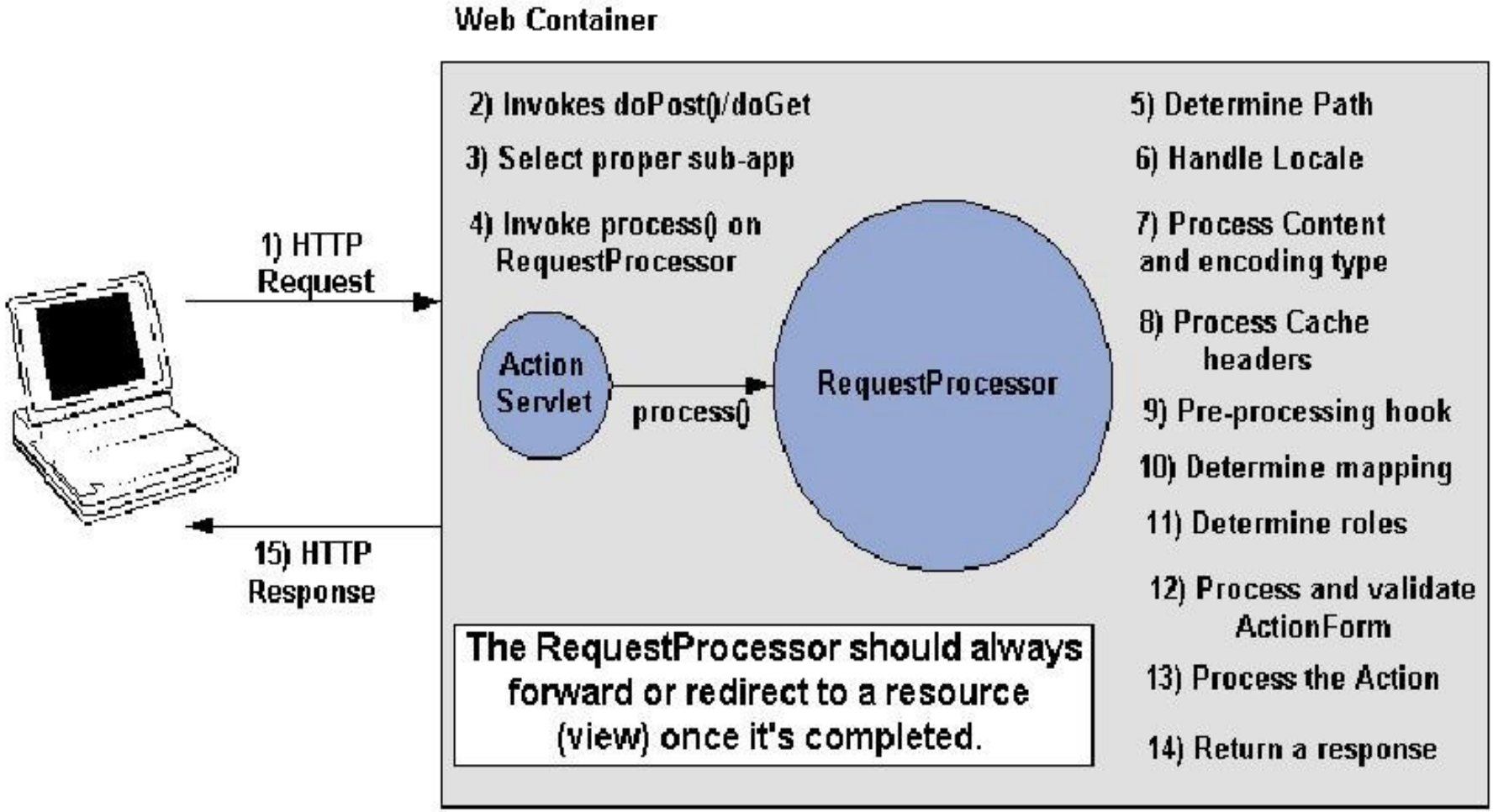
Controller Components

- ActionServlet – (provided by the Framework)
- RequestProcessor – (provided by the Framework)
- Action Classes – (You build these)

The ActionServlet and RequestProcessor

- Receive the HttpServletRequest
- Automatically populate a JavaBean (ActionForm) from the request parameters
- Handle Locale and Content Type Issues
- Based on the URI, select the appropriate Action to handle the request

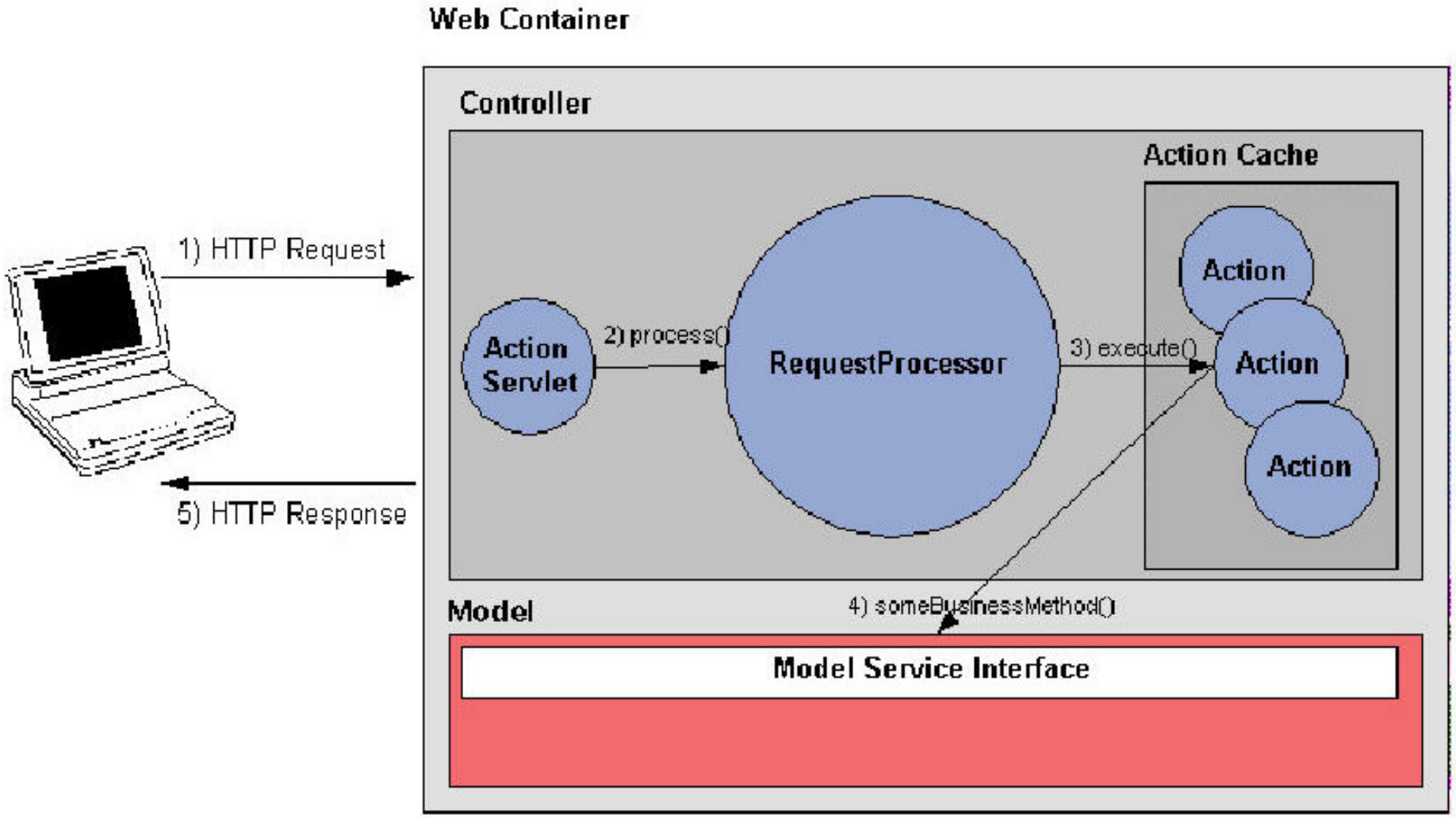
ActionServlet and RequestProcessor



The Action Class

- Extends org.apache.struts.action.Action
- Override the execute() method
- Bridge between the user-invoked URI and the business method residing in the Model class (Command pattern)
- Based on success/failure of processing in Model, determines which view should be rendered next
- Actually part of the Controller, not the Model

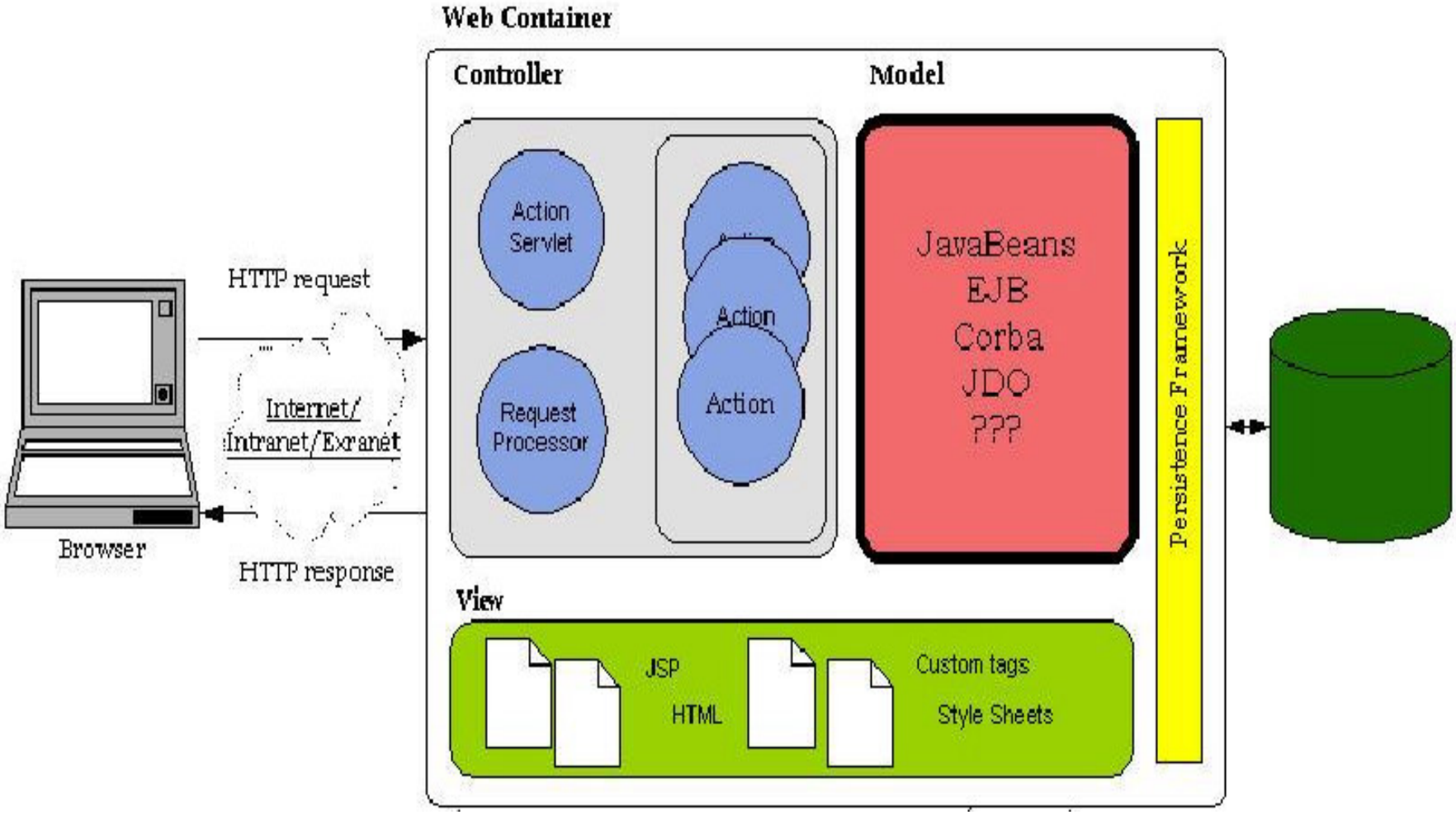
Action Class Diagram



Action Class Example

```
import javax.servlet.http.*;
import org.apache.struts.action.*;
import example.model.*;

public class TransformTextAction extends Action {

    public ActionForward execute(
        ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) throws Exception
    {
        TransformForm myForm = (TransformForm) form;
        String inputText = myForm.getInputText();
        Integer transformType = myForm.getTransformType();
        String resultText =
            TransformModel.transformText(inputText, transformType);
        myForm.reset();
        request.setAttribute("resultText", resultText);

        // Forward control to the specified success URI
        return mapping.findForward("continue");
    }
}
```

The Model Components



Struts Model Components

- No model components provided
- Any component model supported by Struts (JavaBeans, EJB, CORBA, JDO, OJB, etc.)
- Should always attempt to maintain a clean separation from Action and Model

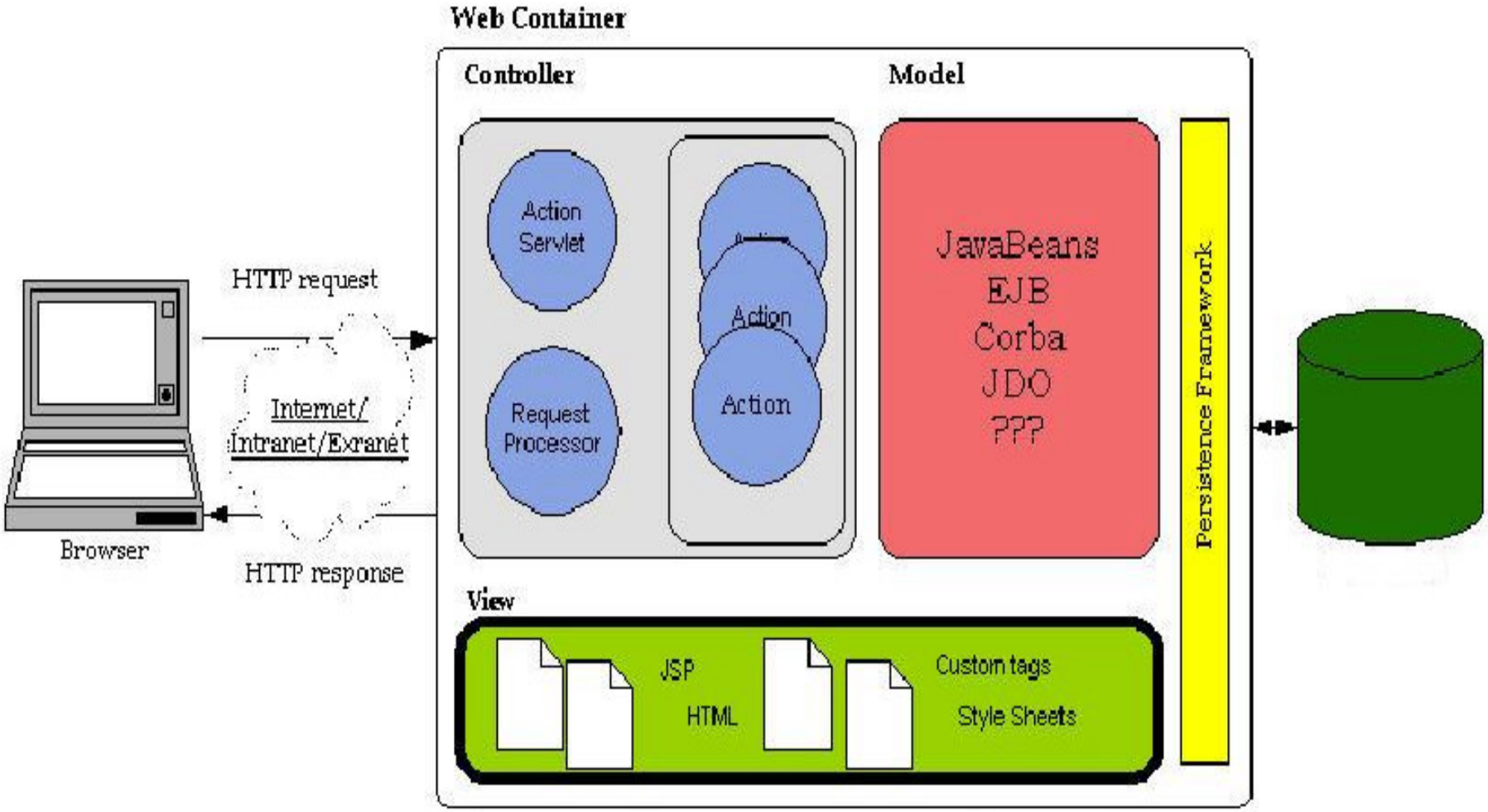
Model Class Example

```
package example.model;

public class TransformModel
{
    public static String transformText(String origText, Integer transType)
    {
        String rc = origText;
        switch(transType.intValue())
        {
        case TransformerConstants.TO_UPPER:
            rc = origText.toUpperCase();
            break;
        case TransformerConstants.TO_LOWER:
            rc = origText.toLowerCase();
            break;
        }
        return rc;
    }
}
```

Model Class Example (cont.)

```
package example.model;

public class TransformerConstants
{
    public static final int TO_UPPER = 0;
    public static final int TO_LOWER = 1;
    public static final int CAP_EACH_WORD = 2;
    public static final int SENTENCE_FORM = 3;
}
```

The View Components



The View Components

- Java Server Pages
- HTML
- JavaScript and Stylesheets
- Multimedia Files
- Resource Bundles
- JavaBeans (Value Objects populated by Model components)
- JSP Custom Tags
- ActionForms

Struts JSP Tag Libraries

- HTML
- Bean
- Logic
- Nested
- Tiles
- Template

HTML Tag Library

- Tags used to create Struts input forms
- Examples include checkbox, image, link, submit, text, and text area

Bean Tag Library

- Tags used for accessing JavaBeans and their properties
- Examples include define, message, write

Logic Tag Library

- Managing conditional generation of output text
- Looping over objects in a collection for repetitive generation of output text
- Application flow management
- Examples include empty, lessThan, greaterThan, redirect, iterate

Bean, Logic, HTML Tag Example

```
<%@ taglib uri="/tags/struts-bean" prefix="bean" %>
<%@ taglib uri="/tags/struts-html" prefix="html" %>
<%@ taglib uri="/tags/struts-logic" prefix="logic" %>
<head>
    <title>Text Transformer</title>
    <html:base/>
</head>
<body bgcolor="white">
  <html:form name="transformForm"
    type="TransformForm" action="transformer.do">
    Please enter the text you would like tranformed:<br/>
    <html:text name="transformForm" property="inputText"/>
    <p/>
    Please select the type of transformation to be performed:<br/>
    <!-- radio buttons -->
    <html:radio name="transformForm" property="transformType" value="0"/>
    UPPER<br/>
    <html:radio name="transformForm" property="transformType" value="1"/>
    lower<br/>
    <logic:present name="resultText" scope="request">
      <b><bean:write name="resultText"/></b>
    </logic:present>
    <p/>
    <html:submit/>
  </html:form>
</body>
```

Sample Application

Text Transformer - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Search Favorites Media

Address http://192.168.40.100:8080/transformer/transformer.jsp

Please enter the text you would like tranformed:

Please select the type of transformation to be performed:

- UPPER
- lower

Submit

Sample Application (cont.)

Text Transformer - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Search Favorites Media

Address http://192.168.40.100:8080/transformer/transformer.jsp

Please enter the text you would like tranformed:

this is a test

Please select the type of transformation to be performed:

- (•) UPPER
- ( ) lower

Submit

Sample Application (cont.)

Text Transformer - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Search Favorites Media

Address http://192.168.40.100:8080/transformer/transformer.do;jsessionid=8E120893E6479DE6972533161B027BE0

Please enter the text you would like tranformed:

Please select the type of transformation to be performed:

- UPPER
- lower

THIS IS A TEST

Submit

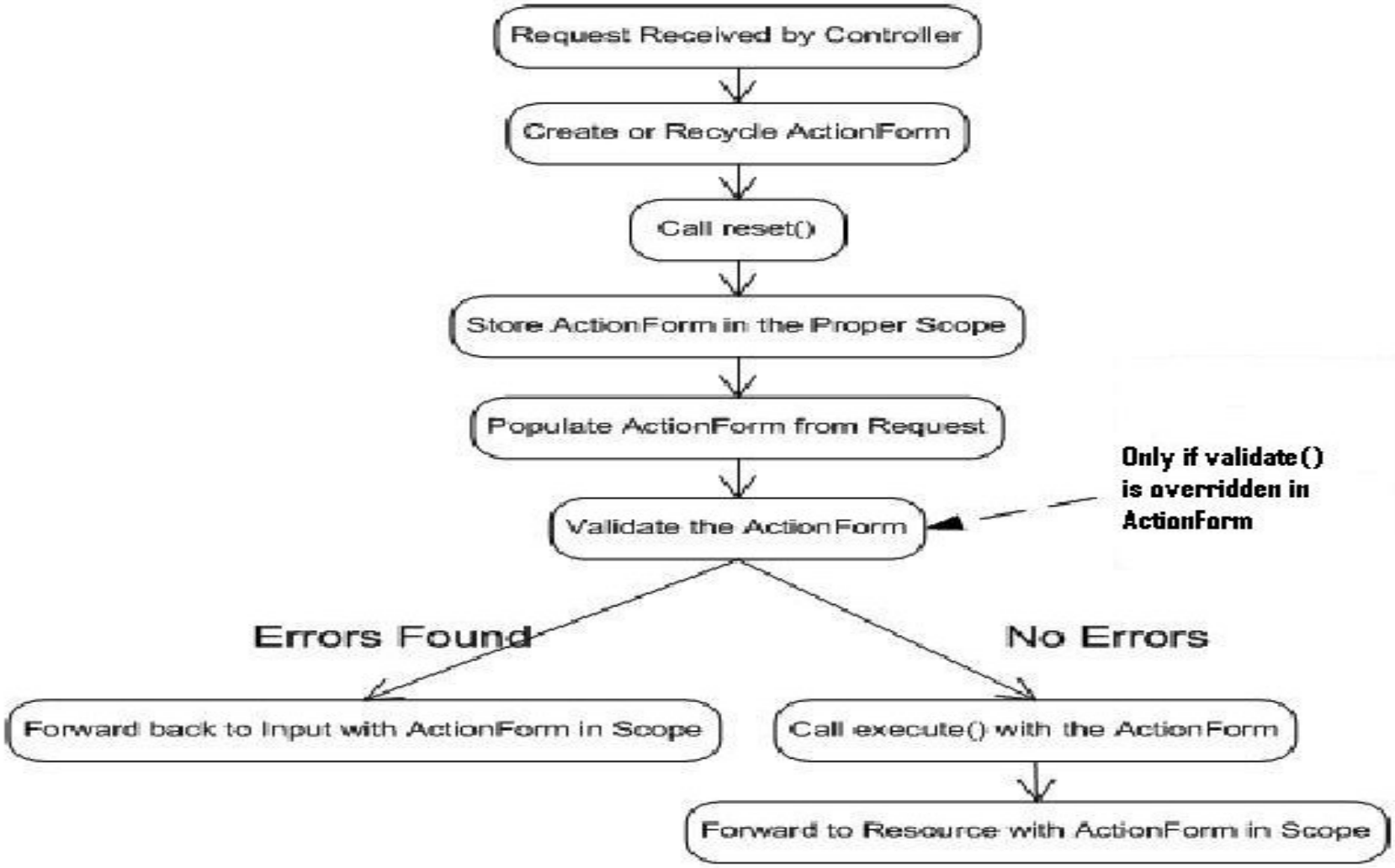
The ActionForm

- Java class that extends org.apache.struts.action.ActionForm
- Captures user data from the HttpRequest
- Stores data temporarily
- Acts as a boundary/go-between between the View and the Controller
- Provides ability to validate the user input

ActionForm Example

```
public class TransformForm
extends org.apache.struts.action.ActionForm {
    private String inputText;
    private Integer transformType;

    public void setInputText(String aInputText) {
        inputText = aInputText;
    }
    public String getInputText() {
        return inputText;
    }
    public void setTransformType(Integer aTransformType) {
        transformType = aTransformType;
    }
    public Integer getTransformType() {
        return transformType;
    }
    public void reset() {
        inputText = "";
        transformType = null;
    }
}
```

ActionForm Sequence of Events



ActionError and ActionMessage

- Used to signify general purpose informational and error messages
- Rely on the ResourceBundles
- JSP Tags have access to them

ActionError and ActionMessage Example

```
<tr class="RED">
  <td>
  </td>
  <td>
    <html:messages id="error">
      <li><bean:write name="error"/></li>
    </html:messages>
  </td>
</tr>
```

Configuring a Struts Application

- Create/edit the web application deployment descriptor (web.xml)
- Create/edit the struts-config.xml file
- Other configuration files as necessary (tiles, validator, etc.)

Configuring the web.xml File

- Add the servlet element
- Configure servlet-mapping element
- Add taglib elements

Sample web.xml

```
<web-app>
  <!-- Standard Action Servlet Configuration (with debugging) -->
  <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>
    <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>

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

Sample web.xml (cont.)

```
  <!-- The Usual Welcome File List -->
  <welcome-file-list>
    <welcome-file>transformer.jsp</welcome-file>
  </welcome-file-list>

  <!-- Struts Tag Library Descriptors -->
  <taglib>
    <taglib-uri>/tags/struts-bean</taglib-uri>
    <taglib-location>/WEB-INF/struts-bean.tld</taglib-location>
  </taglib>

  <taglib>
    <taglib-uri>/tags/struts-html</taglib-uri>
    <taglib-location>/WEB-INF/struts-html.tld</taglib-location>
  </taglib>

  <taglib>
    <taglib-uri>/tags/struts-logic</taglib-uri>
    <taglib-location>/WEB-INF/struts-logic.tld</taglib-location>
  </taglib>
</webapp>
```

The struts-config.xml File

- Uses xml
- Defines the set of “rules” governing a particular Struts application
- As of 1.1, can have multiple configuration files (acting as subordinates to master config file)
- Gets parsed and loaded into memory at startup
- Elements include:
  - Action mappings
  - Form bean definitions
  - Static parts: controller attributes, message resources, plug-in information, and data source definitions

struts-config.xml Example

```
<?xml version="1.0" encoding="ISO-8859-1" ?>

<!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-bean
            name="transformForm"
            type="TransformForm"/>
    </form-beans>
    <action-mappings>
                    <action
                              path="/transformer"
                              type="TransformTextAction"
                              name="transformForm"
                              scope="request"
                              input="transformer.do">
                              <forward name="continue"
                                        path="/transformer.jsp"/>
                    </action>
    </action-mappings>
</struts-config>
```

Internationalization Support

- Much of the framework's functionality is based on java.util.Locale
- Struts uses Java ResourceBundles
- The support from the JDK for normal I18N issues can still be used in a Struts application (date/time formatting, currency formatting/converting, color conventions, etc.)

Packaging and Deployment

- Package as you would any other web application (Web ARchive = WAR file)
- Deploy to any Servlet 2.2/JSP 1.1 compliant container

What to watch for in the future

- Incorporation/closer integration/replacement of Struts tag libraries with those from JSTL
- Java Server Faces – new form of View, allowing more custom component creation on server prior to showing to user

Struts Resources

- Struts home page (http://jakarta.apache.org/struts/)
- Struts user and mailing lists