**Experiment No. 3**

**Title: Write a Struts2 application to implement Internatinalization.**

**Batch:B1 RollNo.:1514033 ExperimentNo.:3**

### Aim: Write a Struts2 application to implement Internatinalization.

### --------------------------------------------------------------------------------------------

**Resources needed:**Eclipse,Tomcat,Struts2.

### Theory:

* *l10n*is the process of creating the actual language-specific texts and formatting; it’s the non programmatic adaptation of the application to another language.
* *i18n*is the process of designing an application so that it has the functionality to change to a different language without resorting to programmatic change of the application; developers need not be involved, technically speaking.

ResourceBundle

Java’s ResourceBundle is an abstract class. It’s up to subclasses to provide an implementation that can manage the resources contained in the bundle, such as the greeting text mentioned earlier. Subclasses of ResourceBundle can manage their resources in any way they like. Greetings could be stored in the database and retrieved with database code contained in the subclass. The Java platform provides a couple of convenient subclasses for your use. The most commonly used of these is the PropertyResourceBundle, which loads its resources from plaintext properties files.Since we’re using properties files to store our resources, we don’t have to write any Java code to implement our bundle. The bundle uses the built-in property Resource-Bundle.

We just provide a set of properties files that follow a specific naming convention, as follows:

BundleName\_languageCode.properties

Defining default resource bundles

* Default ResourceBundles—Struts 2 allows you to specify global bundles that will always be available.
* Set the value for the Struts 2 property

struts.custom.i18n.resources.

* In the struts.properties file or with a constant element in one of your XML configuration files, for example struts.xml

The following shows the constant element, which can be placed in struts.xml

<constant name="struts.custom.i18n.resources" value="global-messages" />

* Using the struts.properties file:

struts.custom.i18n.resources=global-messages

* A comma-separated list of bundles, to be searched in the given order.

struts.custom.i18n.resources=global-messages,manning.utils.otherBundle

getText()

* TextProvidergetText() method is overloaded. Note the following two signatures:
* String getText(String key, List args);
* String getText(String key, String[] args);
* The values in the List or in the String [] are used to fill the parameters in the returned message text.

Working of Struts 2 i18n:

* Make your actions extend ActionSupport, so that they inherit the default TextProvider
* implementation.
* Put some properties files somewhere they can be found by the default TextProvider,
* such as in a properties file witha name mirroring the action.
* Start pulling messages into your pages with the Struts 2 text tag or by hitting
* thegetText() method with direct OGNL.

### Results: (Screen shots of application development steps, program code and web browser displaying the specified message.)

### Code:

### Struts.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE struts PUBLIC "-//Apache Software Foundation//DTD Struts Configuration 2.3//EN"

"http://struts.apache.org/dtds/struts-2.3.dtd">

<struts>

<package name=*"default"* extends=*"struts-default"*>

<action name=*"Instruction"* class=*"com.Action.Instruction"*>

<result name=*"success"*>display.jsp</result>

</action>

<action name=*"jsp\_display"*>

<result>jsp\_display.jsp</result>

</action>

</package>

<constant name=*"struts.custom.i18n.resources"* value=*"com.Action.Lang"* />

### </struts>

### Web.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"http://java.sun.com/xml/ns/javaee"* xsi:schemaLocation=*"http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd"* id=*"WebApp\_ID"* version=*"3.0"*>

<display-name>1514033\_Exp3</display-name>

<welcome-file-list>

<welcome-file>index.html</welcome-file>

<welcome-file>index.htm</welcome-file>

<welcome-file>index.jsp</welcome-file>

<welcome-file>default.html</welcome-file>

<welcome-file>default.htm</welcome-file>

<welcome-file>default.jsp</welcome-file>

</welcome-file-list>

<filter>

<filter-name>struts2</filter-name>

<filter-class>

org.apache.struts2.dispatcher.ng.filter.StrutsPrepareAndExecuteFilter

</filter-class>

</filter>

<filter-mapping>

<filter-name>struts2</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

### </web-app>

### com.Action

### Lang\_en.properties

instruc.message=Twinkle, twinkle, little star,How I wonder what you are.Up above the world so high,Like a diamond in the sky.Twinkle, twinkle, little star,How I wonder what you are!

### Lang\_fr.properties

instruc.message=Brille, brille, petite étoile,Je me demande vraiment ce que tu es!Au-dessus du monde si haut,Comme un diamant dans le ciel.Brille, brille, petite étoile,Je me demande vraiment ce que tu es!

### Instruction.java

**package** com.Action;

**import** com.opensymphony.xwork2.ActionSupport;

**public** **class** Instruction **extends** ActionSupport{

**private** String message;

**public** String execute(){

setMessage(getText("instruc.message"));

**return** "success";

}

**public** String getMessage() {

**return** message;

}

**public** **void** setMessage(String message) {

**this**.message = message;

}

}

### Display.jsp

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<%@ taglib uri=*"/struts-tags"* prefix=*"s"* %>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<s:property value=*"message"*/>

</body>

### </html>

### Index.jsp

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<%@ taglib uri=*"/struts-tags"* prefix=*"s"* %>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<h1>Twinkle Twinkle Little Star Poem</h1>

<s:url id=*"en"* action=*"Instruction"*>

<s:param name=*"request\_locale"*>en</s:param>

</s:url>

<s:a href=*"%{en}"*>English</s:a>

<br>

<s:url id=*"fr"* action=*"Instruction"*>

<s:param name=*"request\_locale"*>fr</s:param>

</s:url>

<s:a href=*"%{fr}"*>French</s:a>

<br>

Using Jsp<br>

<s:url id=*"en"* action=*"jsp\_display"*>

<s:param name=*"request\_locale"*>en</s:param>

</s:url>

<s:a href=*"%{en}"*>English</s:a>

<br>

<s:url id=*"fr"* action=*"jsp\_display"*>

<s:param name=*"request\_locale"*>fr</s:param>

</s:url>

<s:a href=*"%{fr}"*>French</s:a>

</body>

### </html>

### Jsp\_display.jsp

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<%@ taglib uri=*"/struts-tags"* prefix=*"s"* %>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<h2><s:property value=*"getText('instruc.message')"* /></h2>

</body>

### </html>

### Output:

### Home page

### 

### In English

### In french

### 

### In English using jsp

### 

### In French using jsp

### 

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Questions:**

1. Explain Struts2 ResourceBundle location algorithm.

Struts2 ResourceBundle location algorithm :

1. ActionClass- ActionClass\_es.properties

2. MyInterface- MyInterface\_es.properties

3. MySuperClass -MySuperClass\_es.properties

4. If the action implements ModelDriven, theclass of the model object itself will be used to look up ResourceBundles. In other words, if our model object is our User class, then User.properties, and so on

5. package.properties - Next, the search tries to load a package ResourceBundle for the package of the current action class, and every super package back up the chain.

6. In other words, if our current action is in com.action.PortfolioHomepage, the framework will attempt to find a package.properties file located in the com.action package, then in the com package.

7. Domain model object exposed on the ValueStack referenced by key—This is similar to #4, ModelDriven.

### Outcomes: Develop Applications using Frameworks

**Conclusion: (Conclusion to be based on the objectives and outcomesachieved)**

Thus the Internationalization application was implemented and message for each text was displayed successfully.

**Grade: AA / AB / BB / BC / CC / CD/DD**

**Signature of faculty in-charge withdate**

**References:**

1.By James Holmes ; ”Struts: The Complete Reference”; 2nd Edition ; McGraw-Hill

Publication

2.ByChuckCavaness; “Programming Jakarta Struts” ; 2nd Edition ; O'Reilly Media

Publication; 2004