Common-Controls Mehrsprachigkeit

Version 1.6 - Stand: 14. Januar 2006

Common Controls

Herausgeber:
SCC Informationssysteme GmbH
64367 Mühltal

Tel: +49 (0) 6151 / 13 6 31 12
Internet http://www.scc-gmbh.com

Product Site:
http://www.common-controls.com

Inhaltsverzeichnis

1 Einleitung

Dieses Dokument behandelt die Möglichkeiten zur Erstellung mehrsprachiger Anwendungen mit dem Common-Controls Framework.

Die folgende Tabelle gibt zunächst einen Überblick über die betroffenen Kontroll- und Formelemente:

Kontrollelement	Sprachabhängige Komponenten
ListControl	Hauptüberschrift Spaltenüberschriften EmptyText-Attribut
TreeListControl	Hauptüberschrift Spaltenüberschriften EmptyText-Attribut
TabSetControl	Reiter, Tooltips
MenuControl	Haupt- und Untermenüpunkte, Tooltips

Tabelle 1: Überblick Sprachabhängigkeit bei Kontrollelemente

Formelemente	Sprachabhängige Komponenten
Formular	Hauptüberschrift
Formularfelder	Labels, Beschreibungstexte, Tooltips
Schaltflächen	Möglichkeit zur Verwendung sprachabhängiger Images:

Tabelle 2: Überblick Sprachabhängigkeit bei Formularelementen

Zur Internationalisierung einer Anwendung bietet das Common-Controls Frameworks **mehrer Möglichkeiten**, die sich alternativ oder in Kombination einsetzen lassen:

- Konfiguration der Mehrsprachigkeit auf Anwendungsebene.
- Konfiguration der Mehrsprachigkeit auf Sitzungsebene
- Konfiguration der Mehrsprachigkeit innerhalb einer JSP Seite.
- Individuelle Konfiguration der Mehrsprachigkeit für einzelne Kontrollelementen und Formulare

Wenn die Mehrsprachigkeit einmal auf Anwendungsebene festgelegt wurde, muss dies nicht mehr auf der darruntergelegenen Stufe wie einer JSP Seite oder innerhalb eines einzelnen Kontroll- bzw. Formular-elementes erfolgen. Wenn auf einer unteren Ebene dennoch eine andere Sprachbehandlung festgelegt wird, werden automatisch die Einstellungen der übergeordneten Stufe überschrieben. Dies kann dann sinnvoll sein, wenn bestimmte Seiten immer in einer vordefinierten Sprache ausgegeben werden müssen, oder wenn eine Anwendung schrittweise auf Mehrsprachigkeit umgestellt werden soll.

Für die Spracheinstellung gilt die folgende Hierarchie:

- Application Scope
- Session Scope
- Request Scope
- Page Scope
- Kontroll- oder Formularelement Ebene

2 Konfigurationsmöglichkeiten

2.1 Konfiguration der Mehrsprachigkeit auf Anwendungsebene

Die Mehrsprachigkeit einer Anwendung lässt sich auf Anwendungsebene konfigurieren, indem im Servletkontext ein Attribut unter dem Schlüssel **Globals.LOCALENAME_KEY** abgelegt wird. Das Attribut kann mit den folgenden Werten initialisiert werden:

- „true“
  Die Lokalisierung wird aktiviert. Das Framework verwendet das von Struts erzeugte Locale-Objekt für Sprachumsetzungen (→ das Locale Objekt welches in der User Session unter dem Schlüssel **org.apache.struts.Globals.LOCALE_KEY** abgelegt ist).
- „false“
  Die Lokalisierung wird explizit deaktiviert. Damit kann die Einstellung einer höheren Ebene aufgehoben werden
- explizite Angabe einer Locale Id (Bsp.: „de“ oder „en“)
  Alle Sprachumsetzungen erfolgen in der angegebenen Sprache

```
public class FrontController extends ActionServlet {

     /**
      * Constructor for FrontController.
      */
     public FrontController() {
         super();
     }

     /**
      * @see org.apache.struts.action.ActionServlet#init()
      */
     public void init() throws ServletException {

         super.init();

         // Enable resource key translation for all elements
         getServletContext().setAttribute(
             com.cc.framework.Globals.LOCALENAME_KEY, "true");

         // Register all Painter Factories with the favoured GUI-Layout
         // In this case we only use the Default-Layout.
         PainterFactory.registerApplicationPainter(
             getServletContext(), DefPainterFactory.instance());

         PainterFactory.registerApplicationPainter(
             getServletContext(), HtmlPainterFactory.instance());
     }

}
```

CodeSnippet 1: Aktivierung der Mehrsprachigkeit auf Anwendungsebene

Wenn einem Benutzer ein Locale Objekt explizit zugewiesen werden soll, kann hierzu ein entsprechendes Locale Objekt in der Session des Users abgelegt werden. Auf diese Weise lässt sich eine individuelle Sprachunterstützung realisieren. Bei der Anmeldung kann so etwa auf ein vorhandenes Benutzerprofil zurückgegriffen werden.

Die Registrierung des Locale Objektes erfolgt dabei wie folgt:

```
request.getSession().setAttribute(
    org.apache.struts.Globals.LOCALE_KEY,
    java.util.Locale.ENGLISH);
```

2.2 Konfiguration innerhalb einer JSP Seite.

Wenn die Internationalisierung nur für eine einzelne JSP Seiten erfolgen soll, wird der Schlüssel **Globals.LOCALENAME_KEY** mit dem Wert „`true`“ direkt im PageContext der Seite abgelegt. Dabei kann auch gleich eine bestimmte Sprachunterstützung gewählt werden.

```
<% pageContext.setAttribute(com.cc.framework.Globals.LOCALENAME_KEY, "true"); %>
```

oder

```
<% pageContext.setAttribute(com.cc.framework.Globals.LOCALENAME_KEY, "en"); %>
```

Bei der Verwendung von Templates ist zu beachten, dass jedes Template seinen eigenen PageContext erhält und dieser bei Includes nicht weitergegeben wird. In solchen Fällen muss die Einstellung auf allen Template Seiten getrennt erfolgen **oder** die Einstellung erfolgt innerhalb des Haupttemplates, welches die Templates inkludiert, wobei der Schlüssel dann im Request Objekt abgelegt werden sollte. Die Einstellung wird dann gleichermaßen für allen inkludierten Seiten wirksam.

2.3 Konfiguration bei Kontroll- und Formularelementen

Die Spracheinstellung kann auch für jedes Kontroll- oder Formularelement individuell erfolgen. Zur Aktivierung muss das **locale**-Attribute gesetzt werden. Das locale Attribute kann, wie der Schlüssel Globals.LOCALENAME, die folgenden Werte annehmen:

Wert	Bedeutung
`true`	Durch die Angabe von `locale="true"` wird das default Locale Objekt des Users in der Session benutzt (→ unter **org.apache.struts.Globals.LOCALE_KEY** abgelegt).
`locale`	Erlaubt die direkte Angabe einer Locale Id, wie: • locale=”de” • locale=“en“

Tabelle 3: Erlaubte Werte des Locale-Attributes

3 Mehrsprachigkeit in Kontroll- und Formular-elementen

3.1 Kontrollelemente

In Kontrollelementen werden Haupt- und Spaltenüberschriften sowie die Beschriftungen von TabPages innerhalb von TabSets über das **title**-Attribute festgelegt. Der dort angegebene Inhalt wird standardmäßig literal ausgegeben (vgl. Code Snippet 1).

```
<ctrl:list
     action="sample101/userBrowse"
     name="users"
     title="User List"
     rows="10"
     refreshButton="true"
     createButton="true">

     <ctrl:columndrilldown   title="Id"         property="userId"       width="65"/>
     <ctrl:columntext        title="Name"       property="name"         width="350"/>
     <ctrl:columntext        title="Role"       property="role.value"   width="150"/>
     <ctrl:columnedit        title="Edit"/>
     <ctrl:columndelete      title="Delete"/>
</ctrl:list>
```

Code Snippet 1: Kontrollelement ohne Mehrsprachigkeit

Bei einer Internationalisierten Anwendung, wird das **title**-Attribute nicht als Literal angegeben, sondern als Ressourcen Schlüssel. Der Schlüssel wird mit Hilfe der länderspezifischen Application Resource properties Datei in ein konkretes Zeichenkettenliteral übersetzt. Dazu wird der Lokalisierungsmechanismus von Struts genutzt.

```
<ctrl:list
     action="sample101/userBrowse"
     name="users"
     title="userlist1.title"
     rows="10"
     refreshButton="true"
     createButton="true"
     locale="true">

     <ctrl:columndrilldown
             title="userlist1.id"
             property="userId"
             width="65"/>
     <ctrl:columntext
             title="userlist1.name"
             property="name"
             width="350"/>
     <ctrl:columntext
             title="userlist1.role"
             property="role.value"
             width="150"/>
     <ctrl:columnedit
             title="userlist1.edit"/>
     <ctrl:columndelete
             title="userlist1.delete"/>
</ctrl:list>
```

Code Snippet 2: Kontrollelement mit Unterstützung der Mehrsprachigkeit

Zur Unterstützung der verschiedenen Zielsprachen müssen dann in die Ressource Properties Dateien nur die jeweiligen Schlüssel/Werte-Paare eingetragen werden.

Wenn eine Anwendung beispielsweise Englisch und Deutsch als Zielsprachen unterstütz, ergäben sich für das oben gezeigt Code Snippet folgende Einträge:

ApplicationResources_en.properties

```
userlist1.title=User List
userlist1.id=Id
userlist1.name=Name
userlist1.role=Role
userlist1.edit=Edit
userlist1.delete=Delete
```

ApplicationResources_de.properties

```
userlist1.title=Benutzerliste
userlist1.id=Kennung
userlist1.name=Name
userlist1.role=Rolle
userlist1.edit=Bearbeiten
userlist1.delete=Löschen
```

In diesem Beispiel wurde die ApplicationResources.properties Datei in der struts-config.xml als Resource File konfiguriert.

```
<struts-config>

      ...

      <!-- Message Resources (Pakage) -->
      <message-resources parameter="ApplicationResources"/>

</struts-config>
```

3.2 Mehrsprachigkeit in Formularen

Bei einer lokalisierten Anwendung werden die sprachabhängigen Attribute, wie das **caption**-, **label-** oder **title** -Attribute eines Formulars nicht mehr literal angegeben, sondern ebenfalls als Ressourcen Schlüssel. Die Umsetzung in Klartext erfolgt auch hier analog zu den Kontrollelementen.

```
<%@ taglib uri="/WEB-INF/tlds/struts-html.tld"  prefix="html" %>
<%@ taglib uri="/WEB-INF/tlds/cc-base.tld"     prefix="base" %>
<%@ taglib uri="/WEB-INF/tlds/cc-forms.tld"    prefix="forms" %>

<br>

<html:form action="/sample101/userEdit">

    <forms:form
        type="edit"
        caption="frmUserEdit.caption"
        formid="frmEdit"
        locale="true">

        <forms:plaintext
            label="frmUserEdit.id"
            property="userId"/>
        <forms:text
            label="frmUserEdit.lastname"
            property="lastName"
            size="45"
            required="true"/>
        <forms:text
            label="frmUserEdit.firstname"
            property="firstName"
            size="45"
            required="true"/>

        <forms:select
            label="frmUserEdit.role"
            property="rolekey">
            <base:options property="roleOptions"/>
        </forms:select>

        <forms:text
            label="frmUserEdit.email"
            property="email"
            size="45"
            maxlength="256"/>
        <forms:text
            label="frmUserEdit.phone"
            property="phone"
            size="25" />

        <%-- ****************** --%>
        <%-- ** Address      ** --%>
        <%-- ****************** --%>
        <forms:section
            title="frmUserEdit.address">
            <forms:text
                label="frmUserEdit.street"
                property="street"
                size="45" maxlength="80"/>

            <forms:text
                label="frmUserEdit.number"
                property="streetnumber"
                size="5"/>

            <forms:text
                label="frmUserEdit.zipcode"
```

```
                property="zipcode"
                size="5"/>
            <forms:text
                label="frmUserEdit.city"
                property="city"
                size="25"/>

            <forms:select
                label="frmUserEdit.country"
                property="countrycode">
                <base:options property="countryOptions" labelProperty="country"/>
            </forms:select>
        </forms:section>

        <%-- ****************** --%>
        <%-- ** Form Buttons ** --%>
        <%-- ****************** --%>
        <forms:buttonsection default="btnSave">
            <forms:button
                name="btnBack"
                base="images.buttons"
                src="btnBack1.gif"
                title="frmUserEdit.button.cancel"/>
            <forms:button
                name="btnSave"
                base="images.buttons"
                src="btnSave1.gif"
                title="frmUserEdit.button.save"/>
        </forms:buttonsection>
    </forms:form>
</html:form>
```

Für die unterstützten Zielsprachen müssen dann die Schlüssel/Werte-Paare in das Ressource Files der Anwendung eingetragen werden.

Wenn eine Anwendung beispielsweise Englisch und Deutsch als Zielsprachen unterstütz, ergäben sich für das oben gezeigt Code Snippet folgende Einträge:

ApplicationResources_en.properties

```
Images.buttons=app/images/buttons/en

frmUserEdit.caption=User-Edit
frmUserEdit.id=Id
frmUserEdit.name=Name
frmUserEdit.lastname=Last name
frmUserEdit.firstname=First name
frmUserEdit.role=Role
frmUserEdit.email=EMail
frmUserEdit.phone=Phone
frmUserEdit.address=Address
frmUserEdit.street=Street
frmUserEdit.number=Number
frmUserEdit.zipcode=Zipcode
frmUserEdit.city=City
frmUserEdit.country=Country
frmUserEdit.button.cancel=Back
frmUserEdit.button.save=Save
```

ApplicationResources_de.properties

```
Images.buttons=app/images/buttons/de

frmUserEdit.caption=Anwender - Editieren
frmUserEdit.id=Id
frmUserEdit.name=Name
```

```
frmUserEdit.lastname=Nachname
frmUserEdit.firstname=Vorname
frmUserEdit.role=Rolle
frmUserEdit.email=EMail
frmUserEdit.phone=Telefon
frmUserEdit.address=Adresse
frmUserEdit.street=Strasse
frmUserEdit.number=Nummer
frmUserEdit.zipcode=PLZ
frmUserEdit.city=Stadt
frmUserEdit.country=Land
frmUserEdit.button.cancel=Abbrechen
frmUserEdit.button.save=Speichern
```

In diesem Beispiel wurde die ApplicationResources.properties Datei in der struts-config.xml als Resource File konfiguriert.

```
<struts-config>

      ...

      <!-- Message Resources (Pakage) -->
      <message-resources parameter="ApplicationResources"/>

</struts-config>
```

4 Mehrsprachigkeit von Schaltflächen

Um Schaltflächen sprachabhängig zu behandeln wird neben dem **src**-Attribut das **base**-Attribut angegeben. Mit dem base-Attribut wird das Basisverzeichnis für die Grafik angegeben, wobei auch hier wieder der Lokalisierungsmechanismus verwendet werden kann – Das Basisverzeichnis kann also literal oder als Resourcen Schlüssel angegeben werden (in Abhängigkeit von der aktuellen Lokalisierungseinstellung).

Beispiel1 (ohne Base Attribut):

```
<forms:button
    name="btnSave"
    src="app/images/buttons/btnSave1.gif"
    title="frmUserEdit.button.save"/>
```

Es wird das folgende Image verwendet: app/images/buttons/btnSave1.gif

Beispiel2 (Literales Base Attribut):

```
<forms:button
    name="btnSave"
    base="app/images/buttons"
    src="btnSave1.gif"
    title="frmUserEdit.button.save"/>
```

Es wird das folgende Image verwendet: app/images/buttons/btnSave1.gif

Beispiel3 (Lokalisierung mit einem Ressourcen Schlüssel):

```
<forms:button
    name="btnSave"
    base="images.buttons"
    src="btnSave1.gif"
    title="frmUserEdit.button.save"/>
```

Bei den folgenden Ressourcen Einstellungen...

```
ApplicationResources_en.properties

Images.buttons=app/images/buttons/en
```

```
ApplicationResources_de.properties

Images.buttons=app/images/buttons/de
```

… wird das folgende Image verwendet

locale="en" : app/images/buttons/en/btnSave1.gif
locale="de" : app/images/buttons/de/btnSave1.gif

5 Verwendung mehrerer ResourceBundel

Struts erlaubt die Konfiguration eines Resource Bundels in der web.xml oder der struts-config.xml. Durch die Angabe mehrerer **`message-resources`** Tags können Meldungstexte auch aus unterschiedlichen Dateien geladen werden. Das **`key`** Attributes dient hier zur Identifikation des Resource Bundels.

```
<struts-config>

    <!-- Message Resources -->
    <message-resources parameter="ApplicationResources"/>
    <message-resources parameter="MoreApplicationResources"   key="moreResources"/>

</struts-config>
```

Der Zugriff auf eine Ressource aus einem bestimmten Ressource Bundel erfolgt unter Angabe des entsprechenden Schlüssels.

```
<bean:message key="some.message.key" bundle="moreResources"/>
<bean:message key="some.message.key" bundle="moreResources" arg0="1" arg1="2"/>
```

Innerhalb des Common Controls Frameworks erfolgt der Zugriff auf eine Ressource wie folgt:

key@resourcebundel#param1#param2

Wird nur der Schlüssel angegeben wird auf das „default“ ResourceBundel zurückgegriffen, welches unter dem Schlüssel `org.apache.struts.Globals.MESSAGES_KEY` im Servletkontext abgelegt ist.

In dem nachfolgendem Beispiel wird der Tooltip für den Back-Button unter seinem Schlüssel `button.title.back` in der `MoreApplicationResources.properties` Datei gesucht. Der Tooltip für den Save-Button wird hingegen aus der ApplicationResources ermittelt.

```
<forms:buttonsection>
    <forms:button
        base="images.button"
        styleId="btnBack"
        name="btnBack"
        src="btnBack1.gif"
        title="button.title.back@moreResources"/>

<forms:buttonsection>
    <forms:button
        base="images.button"
        styleId="btnSave"
        name="btnSave"
        src="btnSave1.gif"
        title="button.title.save"/>
```

6 Framework Ressource Schlüssel

Das Framework verwendet zur Lokalisierung von internen Meldungstexten, wie etwa in Listen vorbelegten Schlüssel:

Ressource Schlüssel	Message Key	Vorbelegung
ListControl		
FW_ITEMS_NOENTRIES	fw.items.noentries	no entries
FW_ITEMS_1TE	fw.items.1te	1 item
FW_ITEMS_ITEMS	fw.items	{0} items
FW_ITEMS_RANGE	fw.items.range	{0} to {1} of {2}
FW_ITEMS_INFINITE	fw.items.infinite	{0} to {1} of many
FW_EMPTY_TEXT	fw.empty.text	No items in list!!
TreeListControl		
FW_PAGE_NOENTRIES	fw.page.noentries	no entries
FW_PAGE_1TE	fw.page.1te	page 1
FW_PAGE_RANGE	fw.page.range	page {0} of {1}
Tooltips		
FW_TOOLTIP_CHECKALL	fw.tooltip.checkall	check all items
FW_TOOLTIP_CREATE_ITEM	fw.tooltip.create.item	create new item
FW_TOOLTIP_FIRSTPAGE	fw.tooltip.firstpage	goto first page
FW_TOOLTIP_LASTPAGE	fw.tooltip.lastpage	goto last page
FW_TOOLTIP_NEXTPAGE	fw.tooltip.nextpage	goto next page
FW_TOOLTIP_PAGE	fw.tooltip.page	goto page {0}
FW_TOOLTIP_PREVPAGE	fw.tooltip.prevpage	goto previous page
FW_TOOLTIP_REFRESH_LIST	fw.tooltip.refresh.list	refresh list
FW_TOOLTIP_UNCHECKALL	fw.tooltip.uncheckall	uncheck all items
Gauge		
FW_EMPTY_GAUGE	fw.empty.gauge	no element
Tabbset		
FW_TABSET_RANGE	fw.tabset.range	{0} ... {1} from {2}
Tabbar		
FW_TABBAR_RANGE	fw.tabbar.range	{0} ... {1} from {2}
Forms		
FW_FORM_REQUIRED	fw.form.required	required input element
CalendarControl		
FW_CALENDAR_BUTTON_OK_LABEL	fw.calendar.button.ok.label	Ok
FW_CALENDAR_BUTTON_OK_WIDTH	fw.calendar.button.ok.width	80
FW_CALENDAR_BUTTON_OK_TOOLTIP	fw.calendar.button.ok.tooltip	ok
FW_CALENDAR_BUTTON_CANCEL_LABEL	fw.calendar.button.cancel.label	Cancel
FW_CALENDAR_BUTTON_CANCEL_WIDTH	fw.calendar.button.cancel.width	80
FW_CALENDAR_BUTTON_CANCEL_TOOLTIP	fw.calendar.button.cancel.tooltip	cancel
FW_CALENDAR_BUTTON_TODAY_LABEL	fw.calendar.button.today.label	Today
FW_CALENDAR_WINDOW_TITLE	fw.calendar.window.title	DateTimePicker
FW_CALENDAR_WINDOW_WIDTH	fw.calendar.window.width	350
FW_CALENDAR_WINDOW_HEIGHT	fw.calendar.window.height	250
FW_CALENDAR_IMAGE_NEXTMONTH_ALT	fw.calendar.image.nextmonth.alt	

FW_CALENDAR_IMAGE_NEXTMONTH_TOOLTIP	fw.calendar.image.nextmonth.tooltip	next month
FW_CALENDAR_IMAGE_NEXTYEAR_ALT	fw.calendar.image.nextyear.alt	
FW_CALENDAR_IMAGE_NEXTYEAR_TOOLTIP	fw.calendar.image.nextyear.tooltip	next year
FW_CALENDAR_IMAGE_PREVMONTH_ALT	fw.calendar.image.prevmonth.alt	
FW_CALENDAR_IMAGE_PREVMONTH_TOOLTIP	fw.calendar.image.prevmonth.tooltip	prev. month
FW_CALENDAR_IMAGE_PREVYEAR_ALT	fw.calendar.image.prevyear.alt	
FW_CALENDAR_IMAGE_PREVYEAR_TOOLTIP	fw.calendar.image.prevyear.tooltip	prev. year
FW_CALENDAR_MONTHS	fw.calendar.months	
FW_CALENDAR_WEEKDAYS	fw.calendar.weekdays	
HTML_FILE_JSP_CALENDAR	html.file.jsp.calendar	calendar/layout1/calendar.jsp
ColorPicker Control		
FW_COLORPICKER_WINDOW_TITLE	fw.colorpicker.window.title	ColorPicker
Textarea		
FW_TEXTAREA_MAXLENGTH_MESSAGE	fw.textarea.maxlength.message	Characters remaining: {0}/{1}

Die Meldungstexte können lokalisiert werden, indem eine entsprechende properties Datei unter dem Parameter „FrameworkResources“ und dem Schlüssel „com.cc.framework.message“ in der struts.config registriert wird.

Das folgende Beispiel zeigt die Konfiguration von lokalisierten Meldungstexten in der struts.config

Properties Dateien:

FrameworkResources_de.properties
FrameworkResources_en.properties
FrameworkResources_it.properties

Struts.config (Auszug):

```
<!-- Message Resources -->
<message-resources parameter="ApplicationResources"/>
<message-resources parameter="FrameworkResources"      key="com.cc.framework.message"/>
```