

COMMON CONTROLS

Language support

Interface texts, formats and character sets per language

Impressum

PUBLISHED BY

SCC Informationssysteme GmbH
Drosselweg 13
64367 Mühlthal
+49 6151 13 63 10
www.scc-gmbh.com

COPYRIGHT

Copyright © 2000–2026 SCC Informationssysteme GmbH. All rights reserved.

No part of this publication may be stored in a retrieval system, transmitted, or reproduced in any way without the prior agreement and written permission of SCC Informationssysteme GmbH.

TRADEMARKS

Java und JavaServer Pages sind Marken der Oracle Corporation in den USA und anderen Ländern. Microsoft und Microsoft Windows sind eingetragene Marken der Microsoft Corporation in den USA und anderen Ländern. Alle weiteren Produktnamen, Marken, Logos und Symbole können Marken oder eingetragene Marken ihrer jeweiligen Eigentümer sein.

Contents

1 Introduction

2 Ways of configuring

2.1 Configuring language support at application level

2.2 Configuring within a JSP page

2.3 Configuring on control and form elements

3 Language support in control and form elements

3.1 Control elements

3.2 Language support in forms

4 Language support for buttons

5 Using several resource bundles

6 Framework resource keys

1 Introduction

This document covers the ways of building multilingual applications with the Common Controls framework.

The following table gives an overview of the control and form elements concerned:

Control element	Language-dependent components
ListControl	Main heading Column headings emptyText attribute
TreeListControl	Main heading Column headings emptyText attribute
TabSetControl	Tabs, tooltips
MenuControl	Main and submenu items, tooltips

Table 1: Overview of language-dependent parts of the control elements

Form elements	Language-dependent components
Form	Main heading
Form fields	Labels, descriptions, tooltips
Buttons	Language-dependent images can be used:

Table 2: Overview of language-dependent parts of the form elements

The Common Controls framework offers several ways of internationalising an application, which can be used as alternatives or in combination:

- configuring language support at application level;
- configuring language support at session level;
- configuring language support within a JSP page;
- configuring language support individually for single control elements and forms.

Once language support has been set at application level, it need not be set again at a lower level such as a JSP page or a single control or form element. If a lower level does state a different language treatment, it automatically overrides the setting of the level above. That can be useful when certain pages always have to appear in a predefined language, or when an application is to be converted to multiple languages step by step.

The language setting follows this hierarchy:

- application scope
- session scope
- request scope

- page scope
- control or form element level

2 Ways of configuring

2.1 Configuring language support at application level

Language support can be configured at application level by placing an attribute in the servlet context under the key `Globals.LOCALENAME_KEY`. The attribute accepts the following values:

- “true” — localisation is switched on. The framework uses the locale object created by Struts for all language conversions (the locale object held in the user session under the key `org.apache.struts.Globals.LOCALE_KEY`).
- “false” — localisation is explicitly switched off. This cancels the setting made at a higher level.
- an explicit locale id (“de” or “en”, for example) — all language conversions use the language given.

JAVA

```
1 public class FrontController extends ActionServlet {
2
3     /**
4      * Constructor for FrontController.
5      */
6     public FrontController() {
7         super();
8     }
9
10    /**
11     * @see org.apache.struts.action.ActionServlet#init()
12     */
13    public void init() throws ServletException {
14
15        super.init();
16
17        // Enable resource key translation for all elements
18        getServletContext().setAttribute(
19            com.cc.framework.Globals.LOCALENAME_KEY, "true");
20
21        // Register all Painter Factories with the favoured GUI-Layout
22        // In this case we only use the Default-Layout.
23        PainterFactory.registerApplicationPainter(
24            getServletContext(), DefPainterFactory.instance());
25
26        PainterFactory.registerApplicationPainter(
27            getServletContext(), HtmlPainterFactory.instance());
28    }
29
30 }
31
```

Code snippet 1: switching on language support at application level

To assign a locale object to one particular user, place that object in the user's session. This is how per-user language support is achieved: at login, an existing user profile can be drawn upon.

The locale object is registered as follows:

CODE

```

1 request.getSession().setAttribute(
2     org.apache.struts.Globals.LOCALE_KEY,
3     java.util.Locale.ENGLISH);
4

```

2.2 Configuring within a JSP page

If internationalisation is wanted for a single JSP page only, place the key `Globals.LOCALENAME_KEY` with the value `"true"` directly in the page context. A particular language can be chosen at the same time.

CODE

```

1 <% pageContext.setAttribute(com.cc.framework.Globals.LOCALENAME_KEY, "true"); %>
2

```

or

CODE

```

1 <% pageContext.setAttribute(com.cc.framework.Globals.LOCALENAME_KEY, "en"); %>
2

```

When templates are used, bear in mind that every template gets its own page context and that this context is not passed on to included pages. In such cases the setting has to be made separately on every template page — or it is made in the main template that includes the others, in which case the key should be placed in the request object. The setting then applies to all included pages alike.

2.3 Configuring on control and form elements

The language setting can also be made individually for every control or form element by setting the locale attribute. Like the key `Globals.LOCALENAME`, the locale attribute accepts the following values:

Value	Meaning
true	With locale="true" the user's default locale object from the session is used (held under <code>org.apache.struts.Globals.LOCALE_KEY</code>).
locale	Allows a locale id to be given directly, such as: · locale="de" · locale="en"

Table 3: Permitted values of the locale attribute

3 Language support in control and form elements

3.1 Control elements

In control elements the main heading, the column headings and the labels of tab pages within tab sets are set through the title attribute. By default the content given there is output literally (see code snippet 1).

JSP

```
1 <ctrl:list
2     action="sample101/userBrowse"
3     name="users"
4     title="User List"
5     rows="10"
6     refreshButton="true"
7     createButton="true">
8
9     <ctrl:columnmousedown title="Id" property="userId" width="65"/>
10    <ctrl:columnmtext title="Name" property="name" width="350"/>
11    <ctrl:columnmtext title="Role" property="role.value" width="150"/>
12    <ctrl:columnmtext title="Edit"/>
13    <ctrl:columnmdelete title="Delete"/>
14 </ctrl:list>
15
```

Code snippet 1: control element without language support

In an internationalised application the title attribute is not given as a literal but as a resource key. The key is translated into an actual string by means of the country-specific application resource properties file, using the localisation mechanism of Struts.

JSP

```
1 <ctrl:list
2     action="sample101/userBrowse"
3     name="users"
4     title="userlist1.title"
5     rows="10"
6     refreshButton="true"
7     createButton="true"
8     locale="true">
9
10    <ctrl:columnmousedown
11        title="userlist1.id"
12        property="userId"
13        width="65"/>
14    <ctrl:columnmtext
15        title="userlist1.name"
16        property="name"
17        width="350"/>
18    <ctrl:columnmtext
19        title="userlist1.role"
20        property="role.value"
21        width="150"/>
22    <ctrl:columnmtext
23        title="userlist1.edit"/>
24    <ctrl:columnmdelete
```

```
25         title="userlist1.delete"/>
26     </ctrl:list>
27
```

Code snippet 2: control element with language support

To support the various target languages, all that has to be added to the resource properties files are the corresponding key/value pairs.

If an application supports English and German as its target languages, the code snippet above would call for the following entries:

CODE

```
1  ApplicationResources_en.properties
2
3  userlist1.title=User List
4  userlist1.id=Id
5  userlist1.name=Name
6  userlist1.role=Role
7  userlist1.edit=Edit
8  userlist1.delete=Delete
9
10
11 ApplicationResources_de.properties
12
13 userlist1.title=Benutzerliste
14 userlist1.id=Kennung
15 userlist1.name=Name
16 userlist1.role=Rolle
17 userlist1.edit=Bearbeiten
18 userlist1.delete=Löschen
19
```

In this example the file `ApplicationResources.properties` has been configured as the resource file in `struts-config.xml`.

CODE

```
1  <struts-config>
2
3      ...
4
5      <!-- Message Resources (Package) -->
6      <message-resources parameter="ApplicationResources"/>
7
8  </struts-config>
9
```

3.2 Language support in forms

In a localised application the language-dependent attributes of a form — caption, label and title — are no longer given as literals but as resource keys as well. They are turned into plain text just as they are for the control elements.

JSP

```

1  <%@ taglib uri="/WEB-INF/tlds/struts-html.tld" prefix="html" %>
2  <%@ taglib uri="/WEB-INF/tlds/cc-base.tld" prefix="base" %>
3  <%@ taglib uri="/WEB-INF/tlds/cc-forms.tld" prefix="forms" %>
4
5  <br>
6
7  <html:form action="/sample101/userEdit">
8
9      <forms:form
10         type="edit"
11         caption="frmUserEdit.caption"
12         formid="frmEdit"
13         locale="true">
14
15         <forms:plaintext
16             label="frmUserEdit.id"
17             property="userId"/>
18         <forms:text
19             label="frmUserEdit.lastname"
20             property="lastName"
21             size="45"
22             required="true"/>
23         <forms:text
24             label="frmUserEdit.firstname"
25             property="firstName"
26             size="45"
27             required="true"/>
28
29         <forms:select
30             label="frmUserEdit.role"
31             property="rolekey">
32             <base:options property="roleOptions"/>
33         </forms:select>
34
35         <forms:text
36             label="frmUserEdit.email"
37             property="email"
38             size="45"
39             maxlength="256"/>
40         <forms:text
41             label="frmUserEdit.phone"
42             property="phone"
43             size="25" />
44
45         <%-- ***** --%>
46         <%-- ** Address ** --%>
47         <%-- ***** --%>
48         <forms:section
49             title="frmUserEdit.address">
50             <forms:text
51                 label="frmUserEdit.street"
52                 property="street"
53                 size="45" maxlength="80"/>
54
55             <forms:text
56                 label="frmUserEdit.number"
57                 property="streetnumber"
58                 size="5"/>
59
60             <forms:text
61                 label="frmUserEdit.zipcode"
62                 property="zipcode"
63                 size="5"/>

```

```

64         <forms:text
65             label="frmUserEdit.city"
66             property="city"
67             size="25"/>
68
69         <forms:select
70             label="frmUserEdit.country"
71             property="countrycode">
72             <base:options property="countryOptions" labelProperty="country"/>
73         </forms:select>
74     </forms:section>
75
76     <%-- ***** --%>
77     <%-- ** Form Buttons ** --%>
78     <%-- ***** --%>
79     <forms:buttonsection default="btnSave">
80         <forms:button
81             name="btnBack"
82             base="images.buttons"
83             src="btnBack1.gif"
84             title="frmUserEdit.button.cancel"/>
85         <forms:button
86             name="btnSave"
87             base="images.buttons"
88             src="btnSave1.gif"
89             title="frmUserEdit.button.save"/>
90     </forms:buttonsection>
91 </forms:form>
92 </html:form>
93

```

The key/value pairs for the supported target languages then have to be entered into the resource files of the application.

If an application supports English and German as its target languages, the code snippet above would call for the following entries:

CODE

```

1  ApplicationResources_en.properties
2
3  Images.buttons=app/images/buttons/en
4
5  frmUserEdit.caption=User-Edit
6  frmUserEdit.id=Id
7  frmUserEdit.name=Name
8  frmUserEdit.lastname=Last name
9  frmUserEdit.firstname=First name
10 frmUserEdit.role=Role
11 frmUserEdit.email=EMail
12 frmUserEdit.phone=Phone
13 frmUserEdit.address=Address
14 frmUserEdit.street=Street
15 frmUserEdit.number=Number
16 frmUserEdit.zipcode=Zipcode
17 frmUserEdit.city=City
18 frmUserEdit.country=Country
19 frmUserEdit.button.cancel=Back
20 frmUserEdit.button.save=Save
21
22
23 ApplicationResources_de.properties

```

```
24
25 Images.buttons=app/images/buttons/de
26
27 frmUserEdit.caption=Anwender - Editieren
28 frmUserEdit.id=Id
29 frmUserEdit.name=Name
30 frmUserEdit.lastname=Nachname
31 frmUserEdit.firstname=Vorname
32 frmUserEdit.role=Rolle
33 frmUserEdit.email=EMail
34 frmUserEdit.phone=Telefon
35 frmUserEdit.address=Adresse
36 frmUserEdit.street=Strasse
37 frmUserEdit.number=Nummer
38 frmUserEdit.zipcode=PLZ
39 frmUserEdit.city=Stadt
40 frmUserEdit.country=Land
41 frmUserEdit.button.cancel=Abbrechen
42 frmUserEdit.button.save=Speichern
43
```

In this example the file `ApplicationResources.properties` has been configured as the resource file in `struts-config.xml`.

CODE

```
1 <struts-config>
2
3     ...
4
5     <!-- Message Resources (Package) -->
6     <message-resources parameter="ApplicationResources"/>
7
8 </struts-config>
9
```

4 Language support for buttons

To make buttons language-dependent, the base attribute is given alongside the src attribute. The base attribute names the base directory of the graphic, and here too the localisation mechanism can be used: the base directory may be given as a literal or as a resource key, depending on the current localisation setting.

Example 1 (without a base attribute):

JSP

```
1 <forms:button
2     name="btnSave"
3     src="app/images/buttons/btnSave1.gif"
4     title="frmUserEdit.button.save"/>
5
```

The following image is used: app/images/buttons/btnSave1.gif

Example 2 (literal base attribute):

JSP

```
1 <forms:button
2     name="btnSave"
3     base="app/images/buttons"
4     src="btnSave1.gif"
5     title="frmUserEdit.button.save"/>
6
```

The following image is used: app/images/buttons/btnSave1.gif

Example 3 (localisation with a resource key):

JSP

```
1 <forms:button
2     name="btnSave"
3     base="images.buttons"
4     src="btnSave1.gif"
5     title="frmUserEdit.button.save"/>
6
```

With the following resource settings...

CODE

```
1 ApplicationResources_en.properties
2
3 Images.buttons=app/images/buttons/en
4
5
6 ApplicationResources_de.properties
7
8 Images.buttons=app/images/buttons/de
```

... the following image is used:

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

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

5 Using several resource bundles

Struts allows a resource bundle to be configured in web.xml or in struts-config.xml. By giving several message-resources tags, message texts can be loaded from different files; the key attribute identifies the resource bundle.

CODE

```
1 <struts-config>
2
3 <!-- Message Resources -->
4 <message-resources parameter="ApplicationResources"/>
5 <message-resources parameter="MoreApplicationResources" key="moreResources"/>
6
7 </struts-config>
8
```

A resource from a particular resource bundle is accessed by stating the corresponding key.

JSP

```
1 <bean:message key="some.message.key" bundle="moreResources"/>
2 <bean:message key="some.message.key" bundle="moreResources" arg0="1" arg1="2"/>
3
```

Within the Common Controls framework a resource is accessed as follows:

key@resourcebundel#param1#param2

If only the key is given, the default resource bundle is used — the one held in the servlet context under the key org.apache.struts.Globals.MESSAGES_KEY.

In the example below, the tooltip for the back button is looked up under its key button.title.back in the file MoreApplicationResources.properties, whereas the tooltip for the save button is taken from ApplicationResources.

JSP

```
1 <forms:buttonsection>
2 <forms:button
3     base="images.button"
4     styleId="btnBack"
5     name="btnBack"
6     src="btnBack1.gif"
7     title="button.title.back@moreResources"/>
8
9 <forms:buttonsection>
10 <forms:button
11     base="images.button"
12     styleId="btnSave"
13     name="btnSave"
14     src="btnSave1.gif"
15     title="button.title.save"/>
16
```

6 Framework resource keys

For its own internal message texts — the keys preset in lists, for example — the framework uses:

Resource key	Message key	Default
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.prevpag	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		

Resource key	Message key	Default
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.prevmmonth.alt	
FW_CALENDAR_IMAGE_PREVMONTH_TOOLTIP	fw.calendar.image.prevmmonth.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
ColorPickerControl		
FW_COLORPICKER_WINDOW_TITLE	fw.colorpicker.window.title	ColorPicker
Textarea		
FW_TEXTAREA_MAXLENGTH_MESSAGE	fw.textarea.maxlength.message	Characters remaining: {0}/{1}

These message texts can be localised by registering a properties file in struts-config under the parameter "FrameworkResources" and the key "com.cc.framework.message".

The following example shows how localised message texts are configured in struts-config.

Properties files:

FrameworkResources_de.properties

FrameworkResources_en.properties

FrameworkResources_it.properties

struts-config (excerpt):

CODE

```
1 <!-- Message Resources -->
2 <message-resources parameter="ApplicationResources"/>
3 <message-resources parameter="FrameworkResources" key="com.cc.framework.message"/>
4
```