

COMMON CONTROLS · GUIDED TOUR

MenuControl

Adding the control element to an existing application,
step by step

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 Guided Tour MenuControl

1.1 Object

1.2 Registering the painter factory

1.3 Building the horizontal menu in the JSP page

1.4 Building the vertical menu

1.5 Selecting a menu item through the menu context

1.6 Tour end

1.7 Excursus: naming convention for images

2 Glossary

1 Guided Tour MenuControl

1.1 Object

This tour demonstrates the use of the MenuControl. The control element builds the various menus of an application.



Figure 1: Example - horizontal main menu

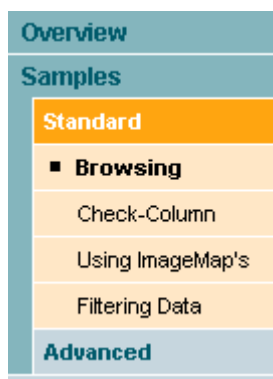


Figure 2: Example - vertical sub-menu

Integrating the MenuControl takes the following steps:

1. Choosing the design of the user interface
2. Defining the menu structure in the JSP page
3. Selecting a menu item at runtime

1.2 Registering the painter factory

The first step is to register the painter factory. It determines the design of the user interface. This can be done for the whole application in the `init()` method of the front controller servlet.¹ Here we choose the standard design provided by the `DefaultPainter`.²

JAVA

```
1  import javax.servlet.ServletException
2
3  import org.apache.struts.action.ActionServlet;
4  import com.cc.framework.ui.painter.PainterFactory
5  import com.cc.framework.ui.painter.def.DefPainterFactory;
6  import com.cc.framework.ui.painter.html.HtmlPainterFactory;
7
8  public class MyFrontController extends ActionServlet {
9
10     public void init() throws ServletException {
11
12         super.init();
13
14         // Register all Painter Factories
15         // with the preferred GUI-Layout
16         // In this case we only use the Default-Layout.
17         PainterFactory.registerApplicationPainter (
18             getServletContext(), DefPainterFactory.instance());
19         PainterFactory.registerApplicationPainter (
20             getServletContext(), HtmlPainterFactory.instance());
21     }
22 }
```

¹ If individual users are to choose between different interface designs, additional painter factories are registered in the user session. This is usually done in the `LoginAction` with `PainterFactory.registerSessionPainter()` in session scope.

² Further designs (painter factories) are part of the Professional Edition, or you develop your own.

1.3 Building the horizontal menu in the JSP page

To use the MenuControl on a JSP page, the corresponding tag library has to be declared at the top of the page. The menu elements are then available with the prefix `<menu:tagname />`. [The tag library also has to be listed in the deployment descriptor, the file `WEB-INF/web.xml`].

In our example we use graphics for the items of the main menu. We therefore start by defining an **image map**, which we assign to the menu through the attribute **imagemap**. The image map holds one image per menu item for each of the states "selected" and "unselected".

To reference the graphic in the image map, every menu item gets an **imageref** attribute that matches the selection rule in the image map. Without an image map, the text recorded for each menu item is shown instead.

JSP

```

1  <util:imagemap name="im_menu">
2      <util:imagemapping rule="tree.unsel" src="images/btnTree1.gif"/>
3      <util:imagemapping rule="tree.sel" src="images/btnTree5.gif"/>
4
5      <util:imagemapping rule="list.unsel" src="images/btnList1.gif"/>
6      <util:imagemapping rule="list.sel" src="images/btnList5.gif"/>
7
8      <util:imagemapping rule="main.unsel" src="images/btnMain1.gif"/>
9      <util:imagemapping rule="main.sel" src="images/btnMain5.gif"/>
10
11     <util:imagemapping rule="about.unsel" src="images/btnAbout1.gif"/>
12     <util:imagemapping rule="about.sel" src="images/btnAbout5.gif"/>
13 </util:imagemap>
14
15 <menu:menu id="main" type="main" imagemap="im_menu">
16
17     <menu:menuitem
18         id="list"
19         text="List Control"
20         imageref="list"
21         action="listcontrol/features.do"/>
22
23     <menu:menuitem
24         id="tree"
25         text="Tree Control"
26         imageref="tree"
27         action="treecontrol/features.do"/>
28
29 </menu:menu>
30
31 <menu:menu id="tools" type="tools" imagemap="im_menu">
32
33     <menu:menuitem
34         id="main"
35         text="Main"
36         imageref="main"
37         action="main.do"/>
38
39     <menu:menuitem
40         id="about"
41         text="About"
42         imageref="about"
43         action="about.do"/>
44
45 </menu:menu>

```

1.4 Building the vertical menu

Unlike the horizontal menu, the vertical menu is not supported with graphical elements by the DefaultPainter. Its items are rendered as text and formatted through style sheets.

A menu item is referenced by a path expression made up of the ids of the individual items along the hierarchy. The menu item "sample101" would be addressed like this: **list/samples/standard/samples101**.

The menu paths have to be defined carefully for the individual menu elements to work together. Since our main menu defines an item with id="list", the matching main menu item is selected at the same time.

JSP

```
1  <%@ taglib uri="/WEB-INF/tlds/cc-menu.tld" prefix="menu" %>
2
3  <menu:menu id="list" type="sidebar" width="135">
4      <menu:menuitem
5          id="overview"
6          text="Overview"
7          action="listcontrol/features.do">
8          <menu:menuitem
9              id="features"
10             text="Features"
11             action="listcontrol/features.do"/>
12      </menu:menuitem>
13      <menu:menuitem
14          id="samples"
15          text="Samples"
16          action="listcontrol/sample101/userBrowse.do">
17          <menu:menuitem
18              id="standard"
19              text="Standard"
20              action="listcontrol/sample101/userBrowse.do">
21              <menu:menuitem
22                  id="sample101"
23                  text="Browsing"
24                  action="listcontrol/sample101/userBrowse.do"/>
25              <menu:menuitem
26                  id="sample102"
27                  text="Check-Column"
28                  action="listcontrol/sample102/carBrowse.do"/>
29              <menu:menuitem
30                  id="sample103"
31                  text="Using ImageMap's"
32                  action="listcontrol/sample103/userRoleBrowse.do"/>
33              <menu:menuitem
34                  id="sample104"
35                  text="Filtering Data"
36                  action="listcontrol/sample104/manufacturerBrowse.do"/>
37          </menu:menuitem>
38          <menu:menuitem
39              id="advanced"
40              text="Advanced"
41              action="listcontrol/sample121/addressBrowse.do">
42              <menu:menuitem
43                  id="sample121"
44                  text="ServerSide DesignModel"
45                  action="listcontrol/sample121/addressBrowse.do"/>
46              <menu:menuitem
47                  id="sample131"
48                  text="Dynamic DataModel"
49                  action="listcontrol/sample131/dynamicListBrowse.do"/>
```

```
50         </menu:menuItem>
51     </menu:menuItem>
52 </menu:menu>
```

1.5 Selecting a menu item through the menu context

A menu item is selected through the menu context, which can be set at runtime in an action or directly in a JSP page. The item is addressed by a path expression. In the example below, the sub-menu defined in the previous chapter is embedded through a template (ListSideBar.jsp). When the JSP page is called, the menu context is set and the item addressed by the path is selected.

JSP

```
1 <%@ taglib uri="/WEB-INF/tlds/cc-template.tld" prefix="template" %>
2 <%@ taglib uri="/WEB-INF/tlds/cc-menu.tld" prefix="menu" %>
3
4 <menu:ctx path="list/samples/standard/sample101"/>
5
6 <template:insert template="/jsp/template/Main.jsp">
7   <template:put
8     name="title"
9     content="User List Common-Controls (Sample101)"
10    direct="true"/>
11
12   <template:put
13     name="header"
14     content="/jsp/template/Header.jsp"/>
15
16   <template:put
17     name="sidebar"
18     content="/jsp/include/ListSideBar.jsp"/>
19
20   <template:put
21     name="about"
22     content="/jsp/list/sample101/About.jsp"/>
23
24   <template:put
25     name="content"
26     content="/jsp/list/sample101/UserBrowseContent.jsp"/>
27
28 </template:insert>
```

As a result, every menu shows those of its items selected that fit the context path completely:

- In the vertical menu the item "list" is shown as selected.
- In the horizontal menu the menu group "list/samples" on level 1, the menu group "list/samples/standard" on level 2 and the item "list/samples/standard/sample101" on level 3 are shown as selected.

1.6 Tour end

The example has shown how little effort it takes to build menus. The DefaultPainter supports vertical sub-menus of up to three levels. Where a deeper navigation structure is needed, the TreeControl is the better choice. Other designs are a matter of adapting the painter or integrating one of your own.

Features of the MenuControl:

- Main and sub-menu items are selected by a path expression through the menu context.
- The main menu can be rendered as text or with graphical elements.
- Supports filters that show menu items depending on the context rather than on permissions.
- Supports showing menu items depending on the user's permissions.
- The design can be adapted to your own style guide (corporate identity) through a painter factory.
- Compact HTML code.
- Same look and feel in Microsoft Internet Explorer > 5.x and Netscape Navigator > 7.x.

1.7 Excursus: naming convention for images

The main menu can render its items as text or as graphics. Graphics can also carry a hover effect. For that, the graphics have to follow the naming convention described below.

Naming convention and states of a menu item:

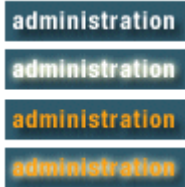


Figure 3: States of a menu item

- Unselected: btnXXX1.gif
- Unselected with hover: btnXXX3.gif
- Selected: btnXXX5.gif
- Selected with hover: btnXXX6.gif

2 Glossary

CC	Common Controls
JSP	JavaServer Pages
Painter	Renders the HTML code of a control element; a custom painter factory adapts the layout to your corporate design.
DataModel	Interface through which a control element obtains its display data.