

# **Common-Controls Guided Tour MenuControl**

Version 1.0.3 - Last changed: 23. November 2004

**Herausgeber:**

SCC Informationssysteme GmbH  
64367 Mühlthal

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

Product Site:  
[www.common-controls.com](http://www.common-controls.com)

Copyright © 2000 - 2003 SCC Informationssysteme GmbH.  
All rights reserved. Published 2003

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.

Java™, JavaServer Pages™ are registered trademarks of Sun Microsystems

Windows® is a registered trademark of Microsoft Corporation.

Netscape™ is a registered trademark of Netscape Communications Corp.

All other product names, marks, logos, and symbols may be trademarks or registered trademarks of their respective owners.

## Table of contents

<b>1</b>	<b>Guided Tour MenuControl</b> .....	<b>1</b>
1.1	Object .....	1
1.2	Registration of the Painterfactory .....	2
1.3	Structure of the horizontal menu in the JSP-Page .....	3
1.4	Structure of the vertical menu .....	4
1.5	Selektion of menu options by the menu context path .....	5
1.6	Tour End.....	6
1.7	Exkurs: Name convention for Images .....	7
<b>2</b>	<b>Glossary</b> .....	<b>8</b>

# 1 Guided Tour MenuControl

## 1.1 Object

This exercise demonstrates the use of the MenuControl. Different menus can be generated within an application with the control element.



Figure 1: Example depiction - Horizontal main menu

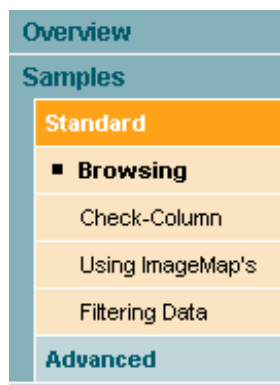


Abbildung 2: Example depiction – Vertikales sub-menu

**The following steps are required for integration of the notwendig:**

1. Selection of the design for the user interface.
2. Defining the menu structure in the JSP-Page.
3. Selection of a menu option at runtime.

## 1.2 Registration of the Painterfactory

The registration of the Painterfactory takes place first. It defines which design the user interface will get. This can be done across the application in the `init()`-method of the `Frontcontroller-Servlet`.<sup>1</sup> Here, we select the standard design that the `DefaultPainter` offers us.<sup>2</sup>

```
import javax.servlet.ServletException;

import org.apache.struts.action.ActionServlet;
import com.cc.framework.ui.painter.PainterFactory;
import com.cc.framework.ui.painter.def.DefPainterFactory;
import com.cc.framework.ui.painter.html.HtmlPainterFactory;

public class MyFrontController extends ActionServlet {

    public void init() throws ServletException {

        super.init();

        // Register all Painter Factories with the preferred GUI-Design
        // In this case we use the Default-Design.
        PainterFactory.registerApplicationPainter (
            getServletContext (), DefPainterFactory.instance());
        PainterFactory.registerApplicationPainter (
            getServletContext (), HtmlPainterFactory.instance());
    }
}
```

---

<sup>1</sup> If it has to be possible for the individual user to choose between different interface designs, then additional `PainterFactory`s are registered in the user session. This is done mostly in the `LoginAction` with `PainterFactory.registerSessionPainter()` in the session Scope.

<sup>2</sup> Additional designs (`PainterFactory`s) are included in the kit of the Professional Edition, or you can develop them yourself.

### 1.3 Structure of the horizontal menu in the JSP-Page

In order to use the MenuControl-tag on a JSP page, the corresponding tag library must be declared at the start of the page. Then, the form elements can be referenced with the prefix `<menu:tagname />`. [In addition, the incorporation of the tag libraries must be included in the Deployment descriptor, the WEB-INF/web.xml file]

In our example, we use graphics to depict the main menu options. Therefore, at the beginning, we define an ImageMap, which we assign to the menu using the attribute **imagemap**. In the ImageMap, for the states "selected" and "unselected", the corresponding relevant images are saved mapped to a menu option. To reference the graphics in the ImageMap, an **imageref** attribute is defined for every menu option, which matches with the selection rule in the ImageMap.

If no ImageMap is specified, the text recorded for the menu options is output.

```
<util:imagemap name="im_menu">
  <util:imagemapping rule="tree.unsel" src="images/btnTree1.gif"/>
  <util:imagemapping rule="tree.sel" src="images/btnTree5.gif"/>

  <util:imagemapping rule="list.unsel" src="images/btnList1.gif"/>
  <util:imagemapping rule="list.sel" src="images/btnList5.gif"/>

  <util:imagemapping rule="main.unsel" src="images/btnMain1.gif"/>
  <util:imagemapping rule="main.sel" src="images/btnMain5.gif"/>

  <util:imagemapping rule="about.unsel" src="images/btnAbout1.gif"/>
  <util:imagemapping rule="about.sel" src="images/btnAbout5.gif"/>
</util:imagemap>

<menu:menu id="main" type="main" imagemap="im_menu">

  <menu:menuitem
    id="list"
    text="List Control"
    imageref="list"
    action="listcontrol/features.do"/>

  <menu:menuitem
    id="tree"
    text="Tree Control"
    imageref="tree"
    action="treecontrol/features.do"/>

</menu:menu>

<menu:menu id="tools" type="tools" imagemap="im_menu">

  <menu:menuitem
    id="main"
    text="Main"
    imageref="main"
    action="main.do"/>

  <menu:menuitem
    id="about"
    text="About"
    imageref="about"
    action="about.do"/>

</menu:menu>
```

## 1.4 Structure of the vertical menu

The default Painter does not support any graphical elements for the vertical menu, in contrast to the horizontal menu. The menu options are output as text and formatted suitably using StyleSheets.

The referencing of a menu option is carried out later, with the help of a path expression, which is composed of the Id's of the individual menu options within the hierarchy. Roughly the following pass specifications would hold good for the menu option "sample101": **list/samples/standard/samples101**.

For the correct interplay between the individual menu elements, particular attention must be paid to defining the menu parts. Since our main menu defines the menu option with the id "list", the corresponding main menu option is selected simultaneously.

```
<%@ taglib uri="/WEB-INF/tlds/cc-menu.tld" prefix="menu" %>

<menu:menu id="list" type="sidebar" width="135">
  <menu:menuitem
    id="overview"
    text="Overview"
    action="listcontrol/features.do">
  <menu:menuitem
    id="features"
    text="Features"
    action="listcontrol/features.do"/>
</menu:menuitem>
  <menu:menuitem
    id="samples"
    text="Samples"
    action="listcontrol/sample101/userBrowse.do">
  <menu:menuitem
    id="standard"
    text="Standard"
    action="listcontrol/sample101/userBrowse.do">
  <menu:menuitem
    id="sample101"
    text="Browsing"
    action="listcontrol/sample101/userBrowse.do"/>
  <menu:menuitem
    id="sample102"
    text="Check-Column"
    action="listcontrol/sample102/carBrowse.do"/>
  <menu:menuitem
    id="sample103"
    text="Using ImageMap's"
    action="listcontrol/sample103/userRoleBrowse.do"/>
  <menu:menuitem
    id="sample104"
    text="Filtering Data"
    action="listcontrol/sample104/manufacturerBrowse.do"/>
</menu:menuitem>
  <menu:menuitem
    id="advanced"
    text="Advanced"
    action="listcontrol/sample121/addressBrowse.do">
  <menu:menuitem
    id="sample121"
    text="ServerSide DesignModel"
    action="listcontrol/sample121/addressBrowse.do"/>
  <menu:menuitem
    id="sample131"
    text="Dynamic DataModel"
    action="listcontrol/sample131/dynamicListBrowse.do"/>
</menu:menuitem>
</menu:menuitem>
</menu:menu>
```

## 1.5 *Selektion of menu options by the menu context path*

The menu context serves to select a menu option, which can be set at runtime in an action or directly in a JSP-Page. The selected menu option is determined by means of a path expression.

In the following example, the sub-menu defined in the previous chapter is integrated through a template (ListSideBar.jsp). If the JSP-Page is called, the menu context is set and the menu option specified via the path is selected.

```
<%@ taglib uri="/WEB-INF/tlds/cc-template.tld" prefix="template" %>
<%@ taglib uri="/WEB-INF/tlds/cc-menu.tld" prefix="menu" %>

<menu:ctx path="list/samples/standard/sample101"/>

<template:insert template="/jsp/template/Main.jsp">
  <template:put
    name="title"
    content="User List Common-Controls (Sample101)"
    direct="true"/>

  <template:put
    name="header"
    content="/jsp/template/Header.jsp"/>

  <template:put
    name="sidebar"
    content="/jsp/include/ListSideBar.jsp"/>

  <template:put
    name="about"
    content="/jsp/list/sample101/About.jsp"/>

  <template:put
    name="content"
    content="/jsp/list/sample101/UserBrowseContent.jsp"/>
</template:insert>
```

The result of this is that in **all** the menus, those menu options are depicted, which completely fit into the context path:

- In the vertical menu, the following menu option is depicted selected:  
Menu option: "list":
- In the horizontal menu, the following main menu options and sub-menu options are depicted selected
  - menu group on level 1: "list/samples"
  - menu group on level 2: "list/samples/standard"
  - the new group on level 3: "list/samples/standard/sample101"



## *1.6 Tour End*

The example has shown how menus can be generated without too much effort. The Default Painter supports vertical sub-menus with up to three levels. If a deeper navigation structure is required, the TreeControl can be used. Other designs can be implemented by customising or integrating your own Painter.





### **Features of the MenuControl:**

- Easy selection of the main menu option and the sub-menu option with the help of a path expression via the menu context.
- Main menu can be depicted as texts or with graphical elements.
- Supports filters for the context-dependent display of many options which are not dependent on authorization.
- Supports the authorization-dependent display of menu options.
- Design through Painterfactory can be matched to own StyleGuide (Corporate Identity).
- Optimized HTML-Code.
- Same Look and Feel in Microsoft InternetExplorer > 5.x and Netscape Navigator > 7.x

## 1.7 Exkurs: Name convention for Images

The main menu offers the option of depicting the menu options as text or graphics. . Hover effects can be used when using graphics. For this, the graphics must follow a name convention as described in the following table.

**Name conventions and states for menu options:**

Sate	Example	Name convention
Unselected		btnXXX1.gif
Unselected HOver		btnXXX3.gif
Selected		btnXXX5.gif
Selected HOver		btnXXX6.gif

The images are automatically exchanged as soon as the mouse pointer is moved over a menu option. A JavaScript Eventhandler, which is registered for the MouseOver and MouseOut-Event in the file fw/def/jscript/controls.js, is responsible for this.

This script is automatically included in every HTML page by the default painter

## **2 Glossary**

**C**

**CC**

Common-Controls