COMMON CONTROLS

Quickstart

Die Common Controls in ein bestehendes Projekt einbauen

SCC Informationssysteme GmbH

Version 1.6 · Stand: 14. Januar 2006

Impressum

HERAUSGEBER

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

Inhalt

1 Quickstart

Dieses Dokument beschreibt die grundlegenden Schritte zum Einsatz der Common-Controls. Die Common-Controls unterstützen Anwendungsentwickler bei der Erzeugung dynamischer HTML Benutzeroberflächen. Dabei wird eine strikte Trennung zwischen der Präsentationsschicht und der Geschäftslogik eingehalten.

1.1 Einrichtung des Projektes

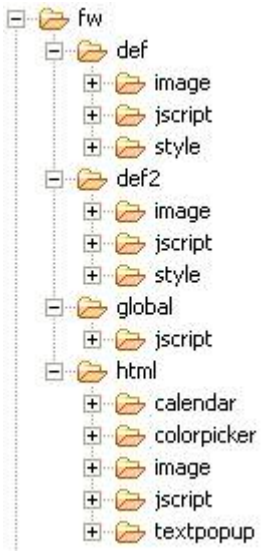
1.1.1 Kopieren der Jar-Files

Für den Einsatz der Common-Controls werden die folgenden JAR-Files in das lib-Verzeichnis des Projektes übernommen:[1]

- antlr.jar
- common-controls-1.x.x.jar
- commons-beanutils.jar
- commons-collections.jar
- commons-digester.jar
- commons-el.jar
- commons-el-ext.jar[2]
- commons-fileupload.jar
- commons-logging.jar
- commons-validator.jar
- ecs.jar
- jakarta-oro.jar
- jakarta-regexp.jar
- log4j.jar
- struts.jar

1.1.2 Kopieren der Resourcen

Für die Darstellung der Benutzeroberfläche werden desweiteren Images und StyleSheets benötigt, die mit den Common-Controls ausgeliefert werden. Hierzu wird das fw-Verzeichnis in das Unterverzeichnis, das als Root-Verzeichnis der Web-Applikation verwendet wird, kopiert.



1.1.3 Kopieren der TagLibrary-Discriptoren (TLD's)

Weiterhin sind folgenden TagLibrary-Discriptoren (TLD's) in das Projekt zu übernehmen:

Common Controls:	Base Elements
cc-base.tld	Kontrollelemente
cc-controls.tld	Typ-Konverter
cc-converter	Formularelemente
cc-forms.tld	Menu
cc-menu.tld	Berechtigungssystem
cc-security.tld	SVG-Tags
cc-svg.tld	JSP-Template
cc-template.tld	Utility
cc-utility.tld	Bean Tags
Struts:	HTMl-Tags
struts-bean	Logic-Tags
struts-html	Nested-Tags
struts-logic	Struts-Tiles
struts-nested	
struts-tiles	

In dem folgenden Kapitel wird davon ausgegangen, dass die tld's in dem Unterverzeichnis /WEB-INF/tlds/ abgelegt werden. Wenn ein anderes Verzeichnis gewählt wird, müssen die nachfolgenden Pfadangaben entsprechend angepasst werden.

Damit die JSP-Tags verwendet werden können, müssen die TagLibrary Discreptoren in dem Deployment-Descriptor der Anwendung (web.xml) registriert werden (siehe Kapitel 1.2).

Hinweis: Ab der JSP-Spezifikation 1.2 ist es auch möglich, direkt auf die TagLibrary Discreptoren welche in eine Jar-Datei gepackt sind zuzugreifen. In diesem Fall kann auf die Registrierung und das Kopieren verzichtet werden.

1.2 Deklaration der TagLibrarys

Um die Common-Control Tags auf einer JSP Seite einzusetzen, muss am Anfang der Seite die entsprechende Tag Library deklariert werden.

Bei einer Registrierung der TLD's im Deployment-Descriptor (web.xml):

JSP

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

oder bei direkter Verwendung der TLD's aus dem common-controls.jar:

JSP

```jsp
<%@ taglib uri="http://www.common-controls.com/cc/tags-ctrl" prefix="ctrl" %>
```

Anschließend können die Common-Controls mit dem Präfix <ctrl:tagname /> referenziert werden. Das Präfix kann frei gewählt werden. Beispiel:

JSP

```jsp
<%@ taglib uri="/WEB-INF/tlds/cc-controls.tld" prefix="ctrl" %>
<%@ taglib uri="/WEB-INF/tlds/cc-utility.tld" prefix="util" %>

<html>
<head>
    <%-- Framework includes --%>
    <util:jsp  directive="includes"/>
</head>

<body leftmargin="0" topmargin="0" onLoad="init();">

<ctrl:tree
    name="products"
    action="sample201/producttreeBrowse"
    root="true"
    linesAtRoot="true"
    labelProperty="name"
    imageProperty="type"
    expandMode="multiple"
    groupselect="true"/>

</body>
</html>

<%-- Framework cleanup processing --%>
<util:jsp  directive="endofpage"/>
```

Jenachdem, welche Funktionalität genutzt wird, müssen eventuell noch weitere Tag Librarys angegeben werden. Welche Tags in welcher TagLibrary verfügbar sind, ist in der Dokumentation der Common Controls TagLibrary beschrieben.

JSP

```jsp
<%@ taglib uri="/WEB-INF/tlds/cc-base.tld"        prefix="base" %>
<%@ taglib uri="/WEB-INF/tlds/cc-controls.tld"    prefix="ctrl" %>
```

```
<%@ taglib uri="/WEB-INF/tlds/cc-converter.tld"   prefix="convert" %>
<%@ taglib uri="/WEB-INF/tlds/cc-forms.tld"       prefix="forms" %>
<%@ taglib uri="/WEB-INF/tlds/cc-menu.tld"        prefix="menu" %>
<%@ taglib uri="/WEB-INF/tlds/cc-security.tld"    prefix="sec" %>
<%@ taglib uri="/WEB-INF/tlds/cc-svg.tld"         prefix="svg" %>
<%@ taglib uri="/WEB-INF/tlds/cc-template.tld"    prefix="template" %>
<%@ taglib uri="/WEB-INF/tlds/cc-utility.tld"     prefix="util" %>
```

Oder:

JSP

```
<%@ taglib uri="http://www.common-controls.com/cc/tags-base"     prefix="base" %>
<%@ taglib uri="http://www.common-controls.com/cc/tags-controls" prefix="ctrl" %>
<%@ taglib uri="http://www.common-controls.com/cc/tags-converter" prefix="convert" %>
<%@ taglib uri="http://www.common-controls.com/cc/tags-forms"    prefix="forms" %>
<%@ taglib uri="http://www.common-controls.com/cc/tags-menu"     prefix="menu" %>
<%@ taglib uri="http://www.common-controls.com/cc/tags-security" prefix="sec" %>
<%@ taglib uri="http://www.common-controls.com/cc/tags-svg"      prefix="svg" %>
<%@ taglib uri="http://www.common-controls.com/cc/tags-template" prefix="template" %>
<%@ taglib uri="http://www.common-controls.com/cc/tags-utility"  prefix="util" %>
```

1.3 Aufnahme der Tag Bibliotheken in den Deployment-Deskriptor

Bei Aufnahme der Tag Bibliotheken in die WEB-INF/web.xml Datei, lässt sich der folgende Abschnitt kopieren:

XML

```
<taglib>
	<taglib-uri>/WEB-INF/tlds/cc-base.tld</taglib-uri>
	<taglib-location>/WEB-INF/tlds/cc-base.tld</taglib-location>
</taglib>
<taglib>
	<taglib-uri>/WEB-INF/tlds/cc-controls.tld</taglib-uri>
	<taglib-location>/WEB-INF/tlds/cc-controls.tld</taglib-location>
</taglib>
<taglib>
	<taglib-uri>/WEB-INF/tlds/cc-converter.tld</taglib-uri>
	<taglib-location>/WEB-INF/tlds/cc-converter.tld</taglib-location>
</taglib>
<taglib>
	<taglib-uri>/WEB-INF/tlds/cc-forms.tld</taglib-uri>
	<taglib-location>/WEB-INF/tlds/cc-forms.tld</taglib-location>
</taglib>
<taglib>
	<taglib-uri>/WEB-INF/tlds/cc-menu.tld</taglib-uri>
	<taglib-location>/WEB-INF/tlds/cc-menu.tld</taglib-location>
</taglib>
<taglib>
	<taglib-uri>/WEB-INF/tlds/cc-security.tld</taglib-uri>
	<taglib-location>/WEB-INF/tlds/cc-security.tld</taglib-location>
</taglib>
<taglib>
	<taglib-uri>/WEB-INF/tlds/cc-svg.tld</taglib-uri>
	<taglib-location>/WEB-INF/tlds/cc-svg.tld</taglib-location>
</taglib>
<taglib>
	<taglib-uri>/WEB-INF/tlds/cc-template.tld</taglib-uri>
	<taglib-location>/WEB-INF/tlds/cc-template.tld</taglib-location>
</taglib>
```

```
<taglib>
	<taglib-uri>/WEB-INF/tlds/cc-utility.tld</taglib-uri>
	<taglib-location>/WEB-INF/tlds/cc-utility.tld</taglib-location>
</taglib>
```

1.4 Registrierung des Resource-Servlets (optional)

Das Resource-Servlet bietet Caching Funktionalitäten und ermöglicht den Zugriff auf Ressourcen (Gif's, SVG-Grafiken), die im ResourceManager registriert sind. Bei einem Relod einer HTML-Seite können durch die Verwendung des Caches Performance Vorteile genutzt werden.

Das Ressource-Servlet wird wie folgt innerhalb der Web.xml (Deployment-Deskriptor) registriert.

JAVA

```
<servlet>
	<servlet-name>res</servlet-name>
	<display-name>Ressource Servlet</display-name>
	<description>Provides access to cached Ressources</description>
	<servlet-class>com.cc.framework.resource.ResourceServlet</servlet-class>
</servlet>

<servlet-mapping>
	<servlet-name>res</servlet-name>
	<url-pattern>*.res</url-pattern>
</servlet-mapping>
```

1.5 Registrierung der Painterfactory

Die PainterFactory kann innerhalb eines Servlets oder mittels eines Struts-Plugin regestriert werden.

1.5.1 Verwendung eines ActionServlets

Die Common-Controls stellen dem Programmierer ein Basisdesign für die Gestaltung der Benutzeroberfläche zur Verfügung, welches nach den eigenen Bedürfnissen erweitert werden kann.[3]

Das Design der Anwendung wird über sogenannte Painterfactories festgelegt. Ein Painter ist dabei für die Erzeugung eines Oberflächendesign verantwortlich. Durch die Verwendung verschiedener Painter, können unterschiedliche Layouts parallel verwendet werden. Die Registrierung der Painterfactory kann zum Beispiel anwendungsweit in der init()-Methode des Frontcontroler-Servlets erfolgen.

JAVA

```
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 the Painter Factory with the prefered GUI-Design
        // In this case we use the Default-Design provided by the DefPainter.
        // If you want to use the Def2-Painter just register
        // the Def2PainterFactory or register your own Painterfactory
        PainterFactory.registerApplicationPainters();
        PainterFactory.registerApplicationPainter (
            getServletContext (), DefPainterFactory.instance());
    }
}
```

Das Actionservlet muss anschließend noch in der web.xml-Datei registriert werden!

Damit sind die grundlegenden Schritte zum Einsatz der Common-Controls abgeschlossen. Der Umgang mit den einzelnen Kontrollelementen wird in den entsprechenden Einzeldokumenten (Guided Tours) beschrieben.

1.5.2 Verwendung eines Struts PlugIn

Bei Verwendung des Struts Frameworks kann Anstelle der Registrierung eines eigenen Servlets auch ein Plugin verwendet werden. Hierzu leite man sich von org.apache.struts.action.PlugIn ab und registriert die PainterFactory in der init()-Methode.

JAVA

```java
import javax.servlet.ServletException;

import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;
import org.apache.struts.action.ActionServlet;
import org.apache.struts.action.PlugIn;
import org.apache.struts.config.ModuleConfig;

import com.cc.framework.ui.painter.PainterFactory;
import com.cc.framework.ui.painter.html.HtmlPainterFactory;
import com.company.framework.ui.painter.app.AppPainterFactory;

/**
 * Struts - Application PlugIn
 */
public class ApplicationPlugin implements PlugIn {

       /**
        * Commons Logging instance.
        */
       private Log log = LogFactory.getLog(this.getClass());

       /**
        * Default Constructor
        */
       public ApplicationPlugin() {
           super();
       }

       /**
        * @see org.apache.struts.action.PlugIn#destroy()
        */
       public void destroy() {
       }

       /**
        * @see org.apache.struts.action.PlugIn#init(ActionServlet, ModuleConfig)
```

```
 */
public void init(ActionServlet actionServlet, ModuleConfig moduleConfig)
        throws ServletException {

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

    // Register the Painter Factory with the prefered GUI-Design
    // In this case we use the Default-Design provided by the DefPainter.
    // If you want to use the Def2-Painter just register
    // the Def2PainterFactory or register your own Painterfactory
    PainterFactory.registerApplicationPainters();
    PainterFactory.registerApplicationPainter (
         actionServlet.getServletContext (), DefPainterFactory.instance());
}
}
```

Das Plugin wird dann noch in der struts-config.xml registriert werden.

XML

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE struts-config PUBLIC "-//Apache Software Foundation//DTD Struts Configuration 1.1//EN"
"http://jakarta.apache.org/struts/dtds/struts-config_1_1.dtd">

<struts-config>

       <!-- Data Sources -->
       <data-sources></data-sources>

       <!-- Form Beans -->
       <form-beans></form-bean>

       </form-beans>

       <!-- Global Exceptions -->
       <global-exceptions></global-exceptions>

       <!-- Global Forwards -->
       <global-forwards></global-forwards>

       <!-- Action Mappings -->
       <action-mappings></action-mappings>

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

       <!-- Application PlugIn -->
       <plug-in className="ApplicationPlugin"/>

</struts-config>

```

1.6 Aufbau der JSP-Seite

Bei Aufbau der JSP-Seite müssen am Anfang und am Ende die in der untenstehenden Abbildung angegebenen Tags aufgenommen werden. Das Framework erhält damit die Möglichkeit eigene Initialisierungen durchzuführen.

Mit dem ersten Tag <util:jsp directive="includes"/> werden alle notwendigen HTML-include Direktiven für das Präsentationsframework eingefügt. Es werden die von den Paintern benötigten StyleSheets und JavaScript Dateien inkludiert.

Das Tag <util:jsp directive="endofpage"/> bewirkt, dass am Ende der JSP Seite alle notwendigen Aufräumarbeiten durchgeführt werden. Dazu zählt beispielsweise das Leeren der Fehlerkollektion.

Um Hover Effekte bei Buttons zu nutzen, muss der JavaScript Handler init() im HTML Body Element aufgenommen werden.

JSP

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

<html>
<head>
	<%-- Framework includes --%>
	<util:jsp  directive="includes"/>
</head>

<body leftmargin="0" topmargin="0" onload="init();">

<!--  Content -->

</body>
</html>

<%-- Framework cleanup processing --%>
<util:jsp  directive="endofpage"/>

```

1.7 Beispiele

Code Beispiele sind in der Demo Applikation enthalten, die auf unserer Homepage zum Download bereitsteht. Dort finden sich auch Konfigurationsbeispiele zu den Kontrollelementen, die verschiedene Verwendungsmöglichkeiten demonstrieren.

2 Anhang Beispiel web.xml

JAVA

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
"http://java.sun.com/dtd/web-app_2_3.dtd">

<web-app id="WebApp">
    <display-name>ccsamples</display-name>
    <description>Sample Application Common-Controls</description>

    <servlet>
        <servlet-name>action</servlet-name>
        <servlet-class>com.cc.sampleapp.servlet.FrontController</servlet-class>

        <!—Struts default modul -->
        <init-param>
            <param-name>config</param-name>
            <param-value>WEB-INF/config/struts-config.xml</param-value>
        </init-param>
        <load-on-startup>1</load-on-startup>
    </servlet>

    <servlet>
        <servlet-name>res</servlet-name>
        <display-name>Ressourcen Servlet</display-name>
        <description>Verwaltung von Ressourcen</description>
        <servlet-class>com.cc.framework.resource.ResourceServlet</servlet-class>
    </servlet>

    <servlet-mapping>
        <servlet-name>action</servlet-name>
        <url-pattern>*.do</url-pattern>
    </servlet-mapping>

    <servlet-mapping>
        <servlet-name>res</servlet-name>
        <url-pattern>*.res</url-pattern>
    </servlet-mapping>

    <session-config>
        <session-timeout>10</session-timeout>
    </session-config>

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

    <!— Common Controls TagLibs-->
    <taglib>
        <taglib-uri>/WEB-INF/tlds/cc-base.tld</taglib-uri>
        <taglib-location>/WEB-INF/tlds/cc-base.tld</taglib-location>
    </taglib>
    <taglib>
        <taglib-uri>/WEB-INF/tlds/cc-controls.tld</taglib-uri>
        <taglib-location>/WEB-INF/tlds/cc-controls.tld</taglib-location>
    </taglib>
    <taglib>
        <taglib-uri>/WEB-INF/tlds/cc-converter.tld</taglib-uri>
```

```
        <taglib-location>/WEB-INF/tlds/cc-converter.tld</taglib-location>
    </taglib>
    <taglib>
        <taglib-uri>/WEB-INF/tlds/cc-forms.tld</taglib-uri>
        <taglib-location>/WEB-INF/tlds/cc-forms.tld</taglib-location>
    </taglib>
    <taglib>
        <taglib-uri>/WEB-INF/tlds/cc-menu.tld</taglib-uri>
        <taglib-location>/WEB-INF/tlds/cc-menu.tld</taglib-location>
    </taglib>
    <taglib>
        <taglib-uri>/WEB-INF/tlds/cc-security.tld</taglib-uri>
        <taglib-location>/WEB-INF/tlds/cc-security.tld</taglib-location>
    </taglib>
    <taglib>
        <taglib-uri>/WEB-INF/tlds/cc-svg.tld</taglib-uri>
        <taglib-location>/WEB-INF/tlds/cc-svg.tld</taglib-location>
    </taglib>
    <taglib>
        <taglib-uri>/WEB-INF/tlds/cc-template.tld</taglib-uri>
        <taglib-location>/WEB-INF/tlds/cc-template.tld</taglib-location>
    </taglib>
    <taglib>
        <taglib-uri>/WEB-INF/tlds/cc-utility.tld</taglib-uri>
        <taglib-location>/WEB-INF/tlds/cc-utility.tld</taglib-location>
    </taglib>

    <!— Struts TagLibs-->
    <taglib>
        <taglib-uri>/WEB-INF/tlds/struts-bean.tld</taglib-uri>
        <taglib-location>/WEB-INF/tlds/struts-bean.tld</taglib-location>
    </taglib>
    <taglib>
        <taglib-uri>/WEB-INF/tlds/struts-html.tld</taglib-uri>
        <taglib-location>/WEB-INF/tlds/struts-html.tld</taglib-location>
    </taglib>
    <taglib>
        <taglib-uri>/WEB-INF/tlds/struts-logic.tld</taglib-uri>
        <taglib-location>/WEB-INF/tlds/struts-logic.tld</taglib-location>
    </taglib>
    <taglib>
        <taglib-uri>/WEB-INF/tlds/struts-nested.tld</taglib-uri>
        <taglib-location>/WEB-INF/tlds/struts-nested.tld</taglib-location>
    </taglib>
    <taglib>
        <taglib-uri>/WEB-INF/tlds/struts-template.tld</taglib-uri>
        <taglib-location>/WEB-INF/tlds/struts-template.tld</taglib-location>
    </taglib>
    <taglib>
        <taglib-uri>/WEB-INF/tlds/struts-tiles.tld</taglib-uri>
        <taglib-location>/WEB-INF/tlds/struts-tiles.tld</taglib-location>
    </taglib>
</web-app>

```

[1] Die Jar-Dateien befinden sich im lib-Verzeichnis des common-controls.zip-Files. Einige jar-Files enthalten im Dateinamen zusätzlich die Versionsnummer.

[2] Das commons-el-ext.jar wird nur dann benötigt, wenn der eingesetzte Applikation-Server über keinen Expression Language (EL) Support verfügt.

[3] Weitere werden mit den nächsten Versionen der Common-Controls geliefert oder können selbst entwickelt werden.