编号：

****

毕业设计(论文)外文翻译

（译文）

院 （系）： **计算机与信息安全学院**

专 业： 计算机科学与技术(卓越工程师)

学生姓名： 严景云

学 号： 1300310226

指导教师单位： 计算机与信息安全学院

姓 名： 汪华登

职 称： 高级实验师

2017年6月5日

目录

[Jakarta Struts Cookbook 1](#_Toc483304879)

[用户界面 1](#_Toc483304880)

[3.1 jstl介绍 1](#_Toc483304881)

[3.1.1 介绍 1](#_Toc483304882)

[3.1.2 使用JSTL 1](#_Toc483304883)

[3.2使用Struts-EL标签 3](#_Toc483304884)

[3.3 显示索引属性 4](#_Toc483304885)

[3.4在表单上使用索引属性 7](#_Toc483304886)

[3.5在JSTL循环中使用索引属性 10](#_Toc483304887)

[3.6从图像提交表单 13](#_Toc483304888)

[3.7在飞行中生成JavaScript 14](#_Toc483304889)

[3.8使用JavaScript动态更改选择选项 15](#_Toc483304890)

# Jakarta Struts Cookbook

# 3 用户界面

## 3.1 jstl引入

### 3.1.1 jstl介绍

面对它:您可以创建最具架构的纯、优雅和健壮的web应用程序，但是如果用户不喜欢他们界面的外观，那么您注定要失败。

对于它：你可以制作出最具有结构化、简单、强壮的web应用程序，但如果用户不喜欢界面的外观，你则注定要失败。 一些Java开发人员认为它优于HTML和JavaScript这样平凡的技术的使用之上。无论你喜欢与否，这些技术的知识，尤其是HTML，在呈现和可用性方面都有不同。如果您不知道如何使用它们来获得您的优势，对于你的应用程序来说，那对你来说将是一个很难得到用户社区认可的挑战。

本章将介绍一些方法，帮助您从应用程序的演示中获得最大的帮助。这里的方法并没有消除一个好的图形/用户界面设计者的需要。但是，它们确实可以帮助您通过Struts的动态功能来影响HTML。此外，本章还将提供基于补充技术的替代解决方案，如JSTL。

使用HTML表单时的一些场景总是具有挑战性。例如，复选框因不检查控件的处理方式而导致令人头疼的问题。这一章包含了一个专门解决这个问题的方法。表单处理中的另一个常见问题是如何处理日期字段。这个有很多方法来做，他们都有自己的优缺点，这一章中也包含了一种强调不同方法的方法。

您还可以找到其他一些解决问题的方法，比如设置表单标签顺序、生成用于JavaScript的url以及使用框架。总而言之，如果你有一个用户界面问题，你很有可能在这些页面中找到一些帮助。

### 3.1.2 使用JSTL

**问题**：

您要在Struts应用程序中使用JSP标准标签库（JSTL）来标记。

**解决:**

从http://jakarta.apache.org/taglibs下载Jakarta Taglibs JSTL参考实现。将lib文件夹中的jstl.jar和standard.jar文件复制到application/WEB-INF/lib文件夹中。 然后将thec.tld，fmt.tld，sql.tld和x.tld文件从tlds文件夹复制到application/WEB-INF文件夹中。

当你想使用jstl，即可在JSP页面中引入适当的taglib指令：<%@ taglib uri="http://java.sun.com/jstl/core" prefix="c" %>

**讨论：**

JSTL是一组功能强大的标记库，应该是任何Struts开发人员工具箱的一部分。JSTL包含用于输出JavaBean属性、循环、条件逻辑和URL格式的标记。有用于格式化和解析日期和数字的标记。XML标记库可以用于在JSP页面上解析和处理XML。SQL标记库的标记与关系数据库进行交互。函数标签库提供了一些有用的函数，这些函数可以用于表达式，主要用于字符串操作。

到目前为止，在您的开发包中，最重要的是核心标记库。这个库包含可以使用的标签，而不是许多Struts bean逻辑标签。为什么要使用这些标签而不是Struts标签呢?答因为这个十分实用性：这些标签比Struts标签更强大，更易于使用。不过，不要搞错，Struts项目的人员不会因此而受到冒犯。恰恰相反。JSTL允许Struts专注于它所做的最好的事情：为健壮的基于jsp的web应用程序提供控制器处理显示。

看看如何使用JSTL标记来实现一个循环，并使用JSTL标记来显示输出。首先，这里是Struts版本:

<ul>

<logic:iterate id="cust" name="branch" property="customers">

<li>

<bean:write name="cust" property="lastName"/>,

<bean:write name="cust" property="firstName"/>

</li>

</logic:iterate>

</ul>

在JSTL标签库中，这个的实现就变得非常简单：

<ul>

<c:forEach var="cust" items="${branch.customers}">

<li>

<c:out value="${cust.lastName}, ${cust.firstName}"/>

</li>

</c:forEach>

</ul>

很酷的一点是，你不需要从另一个中选择一个。当您学习时，JSTL标记可以被引入到一个应用程序中。JSP表达式语言(EL)可以方便地访问各种JSP作用域中可用的动作表单和对象(页面、请求、会话和应用程序)。您要做的最困难的决定不是是否使用JSTL，而是使用JSTL的版本。如果您使用的是JSP 2.0/Servlet 2.4容器，如Tomcat 5，则应该使用JSTL 1.1。否则，您将需要使用JSTL 1.0。

在本书中，在适当的地方，将提供JSTL示例，以及纯粹的基于struts的示例。在许多情况下，提供了使用Struts和JSTL的功能的示例。

看看下面部分：

这个展示了如何使用Struts标签的EL表达式。Hans Bergsten(O'Reilly)的JavaServer Pages详细地介绍了JSTL，是一个非常宝贵的资源。Sun提供了一个关于JSTL的优秀教程，可以在http://java.sun.com/tutorials/jstl中找到。

我为JSTL创建了一个方便的快速参考指南。本指南可以在PDF格式中找到。

## 3.2使用Struts-EL标签

**问题：**  
 您希望能够在Struts标签上使用JSTL表达式来实现属性值获取。  
**解决方案：**  使用在contrib/strutsel/lib目录中提供的Struts发行版的标记库。您需要将该目录中的所有JAR和TLD文件从该目录复制到您的应用程序的WEB-INF / lib目录中，在您想使用表达式的JSP页面上使用适当的atetaglib指令:  
<％@ taglib uri =“http://jakarta.apache.org/struts/tags-html-el” prefix =“html-el”％>  
**讨论**：

jstl风格的表达式，例如$ {foo.bar [4] .baz}，不受基本Struts标签的支持。例如，如果您可以使用如下这样的表达式来格式化一个标记，那将是很不错的选择。

<html：text value =“$ {sessionScope.foo.bar [3]}”/>

相反，这些标签需要运行时表达式，这只是Java代码：

<html:text value="${sessionScope.foo.bar[3]}"/>

从JSP页面中获取Java代码使您的页面不那么脆弱，更易于维护。这种缺乏EL支持的情况得到了普遍支持，并且创建了struts-EL标签库。这些库扩展了html、bean和逻辑Struts标签库，已增加了对EL表达式的支持。如果Struts标签的一个属性支持一个运行时表达式，相应的Struts-EL标签将允许JSTL表达式。它可以在同一个应用程序中使用常规Struts标记和Struts-EL标记，甚至可以在相同的JSP页面上使用，只要确保在每个库的taglib指令中定义唯一的前缀。

但是，对于JSTL来说，struts-EL标签并不是替代品。Struts-EL标签仅为Struts提供了唯一的标记。如果一个Struts标签可以被JSTL标签替换，那么这个标签就不会在Struts-EL标记库中实现。

**放眼JSP 2.0**

如果您正在使用一个支持JSP 2.0的容器，例如Tomcat 5，那么您可以在基于Struts标签中使用EL表达式。然而您并不需要引入struts-EL标签，JSP 2.0支持直接在JSP页面上使用EL表达式。这些表达式可以用于在页面的任何地方呈现动态文本，作为常规HTML标记中的普通文本，以及定制JSP标记的动态属性值。表达式的值是输出，就像使用JSTL c:out标记:

<p>Hello, ${user.firstName}<br />

<html:text value="${sessionScope.foo.bar[3]}"/>

要使用这个功能强大的特性，您必须使用两种方法中的一种来启用EL。如果您的web应用程序使用了Servlet 2.4规范，则默认启用EL。您可以通过查看应用程序的web.xml文件来判断是否正在使用2.4规范。该文件的开头看起来像这样：

<web-app xmlns="http://java.sun.com/xml/ns/j2ee"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.

sun.com/xml/ns/j2ee/web-app\_2\_4.xsd"

version="2.4">

如果您使用的是版本2.3或更低版本，则需要在JSP页面的开头设置以下JSP指令：  
<％@ page isELIgnored =“false”％>

如果使用2.4 DTD，则不需要设置页面指令; EL语法不会被忽略。如果您不确定或有问题，请继续设置页面指令;它不会造成任何问题。

**另请参阅**

详细说明如何配置您的应用程序以使用JSTL

## 3.3 显示索引属性

**问题** 在JSP页面上，您需要从对象的索引属性访问数据。  
**解答**：

使用bean.property [index]来访问索引值，如示例3-1所示。

例3-1。访问索引属性

<@taglib uri = http：//jakarta.apache.org/struts/tags-bean“prefix =”bean“％>

<UL>

<li> <bean：write name =“foo”property =“bar.baz [0]”/> </ li>

<li> <bean：write name =“foo”property =“bar.baz [1]”/> </ li>

<li> <bean：write name =“foo”property =“bar.baz [2]”/> </ li>

</UL>

JSTL支持对索引属性的访问，如示例3-2所示。

示例3-2。访问索引属性（JSTL）

<@taglib uri =“http://java.sun.com/jstl/core”prefix =“c”％>

<UL>

<li> <c：out value =“$ {foo.bar.baz [0]}”/> </ li>

<li> <c：out value =“$ {foo.bar.baz [1]}”/> </ li>

<li> <c：out value =“$ {foo.bar.baz [1]}”/> </ li>

</UL>

**讨论**

索引属性是Struts标记中最容易被误解的方面之一。索引属性是表示一组值的JavaBean属性，而不是单个标量值。使用以下表单的getter方法访问已索引的属性:

public Foo getSomeProperty（int index）{...}

同样，索引属性是使用此形式的setter方法设置的：

public void setFoo（int index，Foo someProperty）{...}

考虑一个表示日历的JavaBean。如示例3-3所示的日CalendarHolder类有一个嵌套的属性，它的monthSet方法用来显示日历中的月份。

实施例3-3。日历JavaBean

package com.oreilly.strutsckbk;

public class CalendarHolder {

private MonthSet monthSet;

public CalendarHolder( ) {

monthSet = new MonthSet( );

}

public MonthSet getMonthSet( ) {

return monthSet;

}

}

MonthSet类，如示例3-4所示，是一个具有索引属性的类，月份表示月份的名称（“1月”、“2月”等等）。  
实施例3-4。具有索引属性的类

package com.oreilly.strutsckbk;

public class MonthSet {

//This document was created by an unregistered ChmMagic, please go to http://www.bisenter.com to register it. Thanks

static String[] months = new String[] {

"January", "February", "March", "April",

"May", "June", "July", "August",

"September", "October", "November", "December"

};

public String[] getMonths( ) {

return months;

}

public String getMonth(int index) {

return months[index];

}

public void setMonth(int index, String value) {

months[index] = value;

}

}

目标是访问JSP页面中CalendarHolder实例的monthSet属性的索引属性月份，如JSP中的以下代码片段所示：

<jsp：useBean id =“calendar”class =“com.oreilly.strutsckbk.CalendarHolder”/>

<UL>

<li> <bean：write name =“calendar”property =“monthSet.month [0]”/> </ li>

<li> <bean：write name =“calendar”property =“monthSet.month [1]”/> </ li>

<li> <bean：write name =“calendar”property =“monthSet.month [2]”/> </ li>

</ UL>

如果要显示的特定索引属性被动态地确定，那么使用的索引是使用JSP脚本变量来设置的，你需要使用scriptlet来生成属性值，如下所示：

您已选择月份号码<bean：write name =“monthIndex”/>：

<bean：write name =“calendar” property ='<％=“monthSet.month [”+ monthIndex +“]”％>“  
 使用scriptlet方法非常难以阅读，甚至更难维护JSP页面。但是，如果您使用JSTL，这将变得更加清洁：

您已选择月份号码<c：out value =“$ {monthIndex}”/>：  
<c：out value =“$ {calendar.monthSet.month [monthIndex]}”/>  
 更常见的是，循环中动态访问索引属性。假设您要显示使用Struts的月份列表  
 逻辑：迭代标签。此标签在集合和数组上进行迭代。以下是在有序列表中显示所有月份的方法：

<ol>

<logic:iterate id="monthName" name="calendar" property="monthSet.months">

<li><bean:write name="monthName"/></li>

</logic:iterate>

</ol>

再次，JSTL可以作为替代。 在JSTL的c：forEach标签比Struts中的logic:iterate标签更容易使用。以下是使用JSTL生成相同有序列表的方法：

<ol>

<c：forEach var =“monthName”items =“$ {calendar.monthSet.months}”>

<li> <c：out name =“$ {monthName}”/> </ li>

</ c：forEach>

</ol>

**另请参阅：**

当您需要使用Struts html标签创建与索引属性对应的表单域时，会出现问题。在目录3.4中已解决了这些特殊问题。在目录3.5中提供了有关在JSTL循环结构中使用索引属性的更多详细信息

## 3.4在表单上使用索引属性

**问题**

您要在与bean的索引属性相对应的表单上创建一组输入字段。

**解答**

在Struts html标签库中的标签上使用indexed属性来生成属性值：

<html:form action =“TestOneAction”>

<logic:iterate name =“MyForm” property =“stringArray” id =“stringValue” indexId =“ctr”>

<BR/>

<html：text property =“stringArray” indexed =“true”/>

</c：foreach>

</ HTML:form>

**讨论**  
 如目录3.3中所示，为显示目的访问索引属性变得很简单。但是，使用表单中的索引属性可能很棘手。如果生成的输入字段的名称格式不正确，则在提交HTML表单时，Struts无法填充ActionForm。 Struts使用Jakarta Commons BeanUtils包从HTTP请求中的值填充ActionForm。具体来说，BeanUtils.populate（）方法从提交表单时发送的HTTP请求数据加载ActionForm。  
 对于索引属性，BeanUtils.populate（）使用请求参数的名称来确定在ActionForm上调用的正确的setter方法。表3-3说明了如何处理不同的表单输入字段名称。该表显示HTML标签，相应的HTTP请求名称/值对，以及处理请求时在ActionForm上调用的方法。  
 考虑一个表单，允许用户输入喜欢的东西，如颜色和网站列表。要保存此数据的ActionForm包含用户​​名称的String属性，表示用户喜爱颜色的aString数组和表示用户喜爱的网站的WebLink对象列表。示例3-5所示的WebLink类定义了一个简单的JavaBean，具有站点名称和URL的属性。  
实例3-5。 WebLink JavaBean

public class WebLink {

public String getName( ) {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getUrl( ) {

return url;

}

public void setUrl(String url) {

this.url = url;

}

private String url;

private String name;

}

现在，您可以创建一个JSP页面（favorites.jsp），允许用户在窗体上输入相应的数据，如示例3-7所示。  
例3-7。 FavoritesForm JSP  
<%@ page contentType="text/html;charset=UTF-8" language="java" %>

<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean" %>

<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html" %>

<%@ taglib uri="/WEB-INF/struts-logic.tld" prefix="logic" %>

<%@ taglib uri="http://java.sun.com/jstl/core" prefix="c" %>

<html:html locale="true">

<head>

<title><bean:message key="index.title"/></title>

<html:base/>

</head>

<body bgcolor="white">

<h2>Favorites Poll</h2>

<html:form action="/admin/ViewFavorites">

<p>

What is your name:<br/>

<html:text property="name"/>

</p>

<p>

What are your three favorite colors:<br/>

<html:text property="color[0]"/><br/>

<html:text property="color[1]"/><br/>

<html:text property="color[2]"/>

</p>

<p>

What are your favorite links?

<table>

<tr>

<th>Name</th>

<th>URL</th>

</tr>

<tr>

<td><html:text property="webLink[0].name"/></td>

<td><html:text property="webLink[0].url"/></td>

</tr>

</table>

</p>

<html:submit/>

<html:reset/>

</html:form>

</body>

</html:html>

由于在示例3-7中的索引值是硬编码的，而不是动态的，所以html:text标记的属性值很容易构造，因此生成的HTML标签具有适当的名称属性值。但是，假设您想要使用这个logic:iterate标签来生成重复的输入字段。为了使用颜色属性，您可能会尝试使用如下的JSP代码：

What are your three favorite colors:

<logic:iterate name="FavoritesForm" id="theColor">

<br/><html:text property="color" indexed="true"/>

</logic:iterate>

这将不会生成所需的HTML标记。索引属性将一个索引(例如1,2,3,..,n)应用到特定Struts html标记的name属性指定的值(在本例中是html:text)。如果您使用上面的代码片段来部署一个JSP，生成的HTML就会像下面这样:

What are your three favorite colors:

<br/><input type="text" name="org.apache.struts.taglib.html.

BEAN[0].color" value="[Ljava.lang.String;@5f1ba8">

<br/><input type="text" name="org.apache.struts.taglib.html.

BEAN[1].color" value="[Ljava.lang.String;@5f1ba8">

<br/><input type="text" name="org.apache.struts.taglib.html.

BEAN[2].color" value="[Ljava.lang.String;@5f1ba8">

该指数不适用于该属性的值。相反，该值应用于表单bean的内部Struts名。另外，value属性包含在数组中调用toString()的结果，而不是数组中的特定元素。

从长远来看，这意味着当您需要设置一个复杂的对象的索引属性的嵌套属性时，索引属性是有用的。一个JavaBean。您可以在逻辑中为非嵌套属性生成输入logic:iterate标签，但是您必须使用scriptlet来生成数组索引:

从长远来看，这意味着，当您需要设置一个复杂对象，即JavaBean的索引属性的嵌套简单属性时，索引属性很有用。您可以在逻辑：iterate标签中为非嵌套属性生成输入字段，但您必须诉诸于scriptlet才能生成数组索引：

What are your three favorite colors:

<logic:iterate name="FavoritesForm" id="theColor" indexId="ctr">

<br/><html:text property='<%="color["+ctr+"]"%>'/>

</logic:iterate>

假设您想要使用这个logic:iterate标签来为最喜欢的链接生成输入字段(Web链接对象)。在这种情况下，索引属性将会执行您想要的内容:

What are your favorite links?

<table>

<tr>

<th>Name</th>

<th>URL</th>

</tr>

<logic:iterate id="webLink" name="FavoritesForm" property="webLinks">

<tr>

<td><html:text name="webLink" property="name" indexed="true"/></td>

<td><html:text name="webLink" property="url" indexed="true"/></td>

</tr>

</logic:iterate>

</table>

使用html标签库的索引属性可能会令人迷惑，这种混乱通常源于name属性的新重要性。在大多数情况下，当使用html标记时，可以忽略该属性的值，因为该值将基于为动作映射声明的表单bean。然而，当使用索引属性时，name属性引用对应的ActionForm的嵌套的索引属性。

回到使用颜色属性的问题，除了使用scriptlet，您还有其他选择。您可以使用strutshtml-el标记或JSTL。这两个选项基本上都与scriptlet相同，但是它们使用EL(表达式语言)，最干净的方法使用了html-el标签:

What are your three favorite colors:

<logic:iterate name="FavoritesForm" id="theColor" indexId="ctr">

<br/><html-el:text property='color[${ctr}]>'/>

</logic:iterate>

如果您更喜欢JSTL，您可以直接生成所需的输入标记，而不是使用Strutshtml标签:

What are your three favorite colors:

<logic:iterate id="color" name="FavoritesForm" property="color" indexId="ctr">

<br/><input type="text" name="color[<c:out value='${ctr}'/>]"

This document was created by an unregistered ChmMagic, please go to http://www.bisenter.com to regis.ter it. Thanks

value="<c:out value='${FavoritesForm.color[ctr]}'/>"/>

</logic:iterate>

这个JSTL版本与使用scriptlet的原始版本一样丑陋。此外，由于Struts html:text和html-el:text标签都不使用，所以html input标记的值属性必须被显式地编码。如果您使用Struts html:text标记，那么该值将自动设置。

**另请参阅**

讨论从表单外部的索引属性显示值时使用的技术。在目录3.5中讨论了如何在JSTL的c:forEach循环中使用index索引属性。

Jakarta Commons项目的BeanUtils包定义了如何解析索引属性。包描述提供了额外的细节，并且可以在这里找到。

## 3.5在JSTL循环中使用索引属性

**问题**

你希望在JSTL的c:forEach循环中的Struts html标签来使用indexed bean属性，而不是Struts的logic:iterate循环。

**解答**

要为一个简单的索引属性创建字段，使用bean:define标签来将循环计数器显示为一个脚本变量，该变量可以在运行时表达式中使用:

<c:forEach var="theItem" items="${MyForm.myItems}" varStatus="loopStatus">

<bean:define id="itemIndex">

<c:out value="${loopStatus.index}"/>

</bean:define>

<br/><html:text property='<%="myItem["+itemIndex+"]"%>'/>

</c:forEach>

如果索引属性是一个嵌套的bean，并且您正在使用indexed=“true”属性，那么就使用JSTL的c:forEach迭代标签来替换Struts的logic:iterate标签。

<c:forEach var="theNestedItem" items="${MyForm.myNestedItems}">

<br/><html:text name="theNestedItem" property="nestedProperty" indexed="true"/>

</c:forEach>

**讨论**

JSTL提供的c:forEach标签提供了额外的功能，并且可以比logic:iterate标签更容易使用。循环遍历的项目可以使用EL进行指定。JSTL标记允许对集合的子集进行更大的控制，并且可以很容易地获得关于循环状态的详细信息。然而，正如所有JSTL标记一样，没有创建脚本变量。如本章其他方法所示，在处理索引属性时，可能需要使用运行时表达式。如果您不使用struts-el标签库，那么这一点尤其正确。

bean：define标签可以从jstl创建的范围变量中创建一个脚本变量。 这个bean：define标签创建一个新的范围变量和一个对应的脚本变量，它基于从值属性或标记体中获取的值。后一种功能提供了JSTL和Struts标记之间的一个有用的桥梁。在解决方案中，bean:define标签用于创建一个变量，其中包含用于访问索引属性的索引。您可以将此技术应用到表单中。

What are your three favorite colors:

<c:forEach var="theColor" items="${FavoritesForm.color}" varStatus="loopStatus">

<bean:define id="ctr">

<c:out value="${loopStatus.index}"/>

</bean:define>

<br/><html:text property='<%="color["+ctr+"]"%>'/>

</c:forEach>

如目录3.4所示，您可以使用Struts-El标签来完全消除scriptlet：

What are your three favorite colors:

<c:forEach var="theColor" items="${FavoritesForm.color}" varStatus="loopStatus">

<br/><html-el:text property='color[${ctr}]'/>

</c:forEach>

如果需要为对象的嵌套属性创建一个HTML输入字段，该属性是一个索引属性，那么在Struts HTML标记中指定indexed =“true”属性。在JSTL的c:forEach循环中使用索引属性的用法与使用logic：iterate标签是相同的。这里有一个例子，在目录3-4中如何说明“最喜欢的链接”部分是怎么做的:

What are your favorite links?

<table>

<tr>

<th>Name</th>

<th>URL</th>

</tr>

<c:forEach var="webLink" items="${FavoritesForm.webLinks}">

<tr>

<td>

<html:text name="webLink" property="name" indexed="true"/>

</td>

<td>

html:text name="webLink" property="url" indexed="true"/>

</td>

</tr>

</c:forEach>

</table>

即使使用begin、end和step属性来控制循环，也可以正确地生成所呈现的索引值。下面使用c:forEach标记演示如何为集合的第一个和第三个元素生成输入字段:

<c:forEach var="webLink" items="${FavoritesForm.webLinks}" begin="1" end="3" step="2">

<tr>

<td>

This document was created by an unregistered ChmMagic, please go to http://www.bisenter.com to regis.ter it. Thanks

<html:text name="webLink" property="name" indexed="true"/>

</td>

<td>

<html:text name="webLink" property="url" indexed="true"/>

</td>

</tr>

</c:forEach>

这将导致以下生成的HTML：

<tr>

<td><input type="text" name="webLink[1].name" value=""></td>

<td><input type="text" name="webLink[1].url" value=""></td>

</tr>

<tr>

<td><input type="text" name="webLink[3].name" value=""></td>

<td><input type="text" name="webLink[3].url" value=""></td>

</tr>

为了在一个循环中显示动态数据，JSTL工作得很好，应该比相应的Struts标记更容易使用。正如您所看到的，JSTL对访问索引属性的支持比Struts标记更好。例如，下面展示了如何显示最喜欢的颜色:

<c：forEach var =“color”items =“$ {favs.color}”>

<li> <c：out value =“$ {color}”/> </li>

</c：forEach>

**另请参阅**

演示了索引属性的相似用法。如果需要，您应该咨询JSTL文档。该规范提供了这里提到的标签的其他细节。该规范可以在http://java.sun.com/jsp/jstl找到。

## 3.6从图像提交表单

**问题**

您希望用户能够通过单击不包含在HTML表单标记中的图像来提交表单。

**解答**

使用与JavaScript网址的链接提交表单：

<html:link href =“javascript：document.MyForm.submit()”>

<html：img page =“/submit-form.gif” alt =“submit”border =“0”/>

</html:link>

**讨论**

Web站点经常使用可点击的图像而不是HTML提交按钮来触发表单提交。Struts 的html:image标签可以用来生成一个html的<input type=“image”/>标签，它可以创建这样的图像。然而，在复杂的HTML布局中，并不总是可以将图像嵌套在<form>… </form>标签中。在某些情况下，HTML页面可以在页面的一个部分中具有多个表单，图像可以在页面的单独区域中提交表单。

上述解决方案可用于从表单标记外部的图像提交表单。要显示的图像嵌套在html:link标签中，该链接通过执行一行JavaScript来提交表单。在解决方案中，JavaScript将提交名为MyForm的表单。表单名称必须与thestruts-config.xml中的相应action元素的name属性相匹配。下面是使用该解决方案生成的HTML:

<a href="javascript:document.MyForm.submit( )">

<img src =“/myApp / struts-power.gif” border =“0”alt =“submit”>

</a>

虽然您可以直接使用上面的HTML标签而不是Struts HTML标签，但是您将失去这些标记提供的特性。通过使用Struts标记，您不必指定上下文名称，并且您可以从资源包中检索图像名和替换文本(如果需要的话)。

另一个选择是使用html:img标签的onclick属性：

<html:img page =“/ submit-form.gif” onclick =“document.MyForm.submit（）;” alt =“提交”border =“0”/>

这种方法的缺点是大多数浏览器都不能提供图像是可点击的视觉线索。由于映像是嵌套在一个链接中，所以大多数浏览器都会更改鼠标指针，以指示可以单击该图像所显示的图像。

**另请参阅**

描述如何向表单的action属性中指定的另一个URL提交表单

## 3.7快速生成JavaScript

**问题**

您希望使用从应用程序的Model中检索的数据来动态生成JavaScript。  
**解答**

使用Struts标记在JavaScript代码中呈现数据，就像您在HTML中所做的那样:

<script language =“JavaScript”>

function showMessage(){

alert（"Hello，<bean：write name ='myForm'property ='name'/>！"）;

}

</script>

**讨论**

上面的解决方案生成一个JavaScript函数，弹出一个包含文本“Hello，name”的消息框,名称的值是用bean: write标签生成的。该解决方案展示了如何使用Struts标签来创建JavaScript，就像它们创建HTML一样简单。

虽然这个解决方案看起来很明显，但是这个问题出现的次数是令人惊讶的。通常的问题是:“我如何从Struts中调用JavaScript函数?”。从技术上讲，您不能在Struts的HTML页面上调用JavaScript函数。Struts和底层JSP技术在服务器端运行。相反，JavaScript是由客户端上的浏览器处理的。然而，这个方法的另一个重要的概念是JSP转换过程。JSP页面由JSP声明、标准JSP标签(如JSP:useBean)、自定义JSP标签(如Struts和JSTL标签)、运行时表达式和scriptlet组成。页面中的其他内容都是模板文本。模板文本是没有作为JSP转换的一部分处理的任何内容。人们通常认为模板文本是HTML标签，但它是JavaScript或其他非jsp处理文本。JSP转换器不关心模板文本的形式。因此，您可以很容易地将文本生成到JavaScript函数中，就像在解决方案中所显示的那样，可以将Struts的动态生成能力作为一种动态生成功能，您可以近似这种行为。

这个解决方案给出了一个简单的场景。如果访问的模型数据需要使用复杂的JavaScript数据结构，比如数组，您可以使用诸如logic:iterate和c:forEach来填充这些结构。

**另请参阅：**

使用迭代标记来生成客户端JavaScript数组

## 3.8使用JavaScript动态更改选择选项

**问题**

您希望使用JavaScript根据从应用程序的模型中检索到的数据动态地更改HTML select元素中显示的项目。

**解答**

使用Struts的logic：iterate标签为不同的选项集创建JavaScript数组，然后使用一个JavaScript的onchange事件处理程序来更改运行时设置的选项。示例3-8展示了一个完整的JSP页面，其中使用Struts标记动态地创建了JavaScript数组。更改选项事件处理程序函数使用JavaScript数组重新设置select控件的选项。

**示例3-8。使用Struts生成DHTML**

<%@ page contentType="text/html;charset=UTF-8" language="java" %>

<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean" %>

<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html" %>

<%@ taglib uri="/WEB-INF/struts-logic.tld" prefix="logic" %>

<html>

<head>

<title>Struts - JavaScript Example</title>

<script language="JavaScript">

// Create the array for the first set of options

fooArray = new Array( );

<logic:iterate id="fooValue" indexId="ctr" name="MyForm" property="fooList">

fooArray[<bean:write name="ctr"/>] = new Option("<bean:write name='fooValue'/>", "<bean:write name='fooValue'/>", false, false);

</logic:iterate>

// Create the array for the second set of options

barArray = new Array( );

<logic:iterate id="barValue" indexId="ctr" name="MyForm" property="barList">

fooArray[<bean:write name="ctr"/>] = new Option("<bean:write name='barValue'/>", "<bean:write name='barValue'/>", false, false);

</logic:iterate>

function changeOptions(var control) {

// control is the triggering control

// baz is the select control

baz = document.MyForm.baz;

baz.options.length=0;

if (control.value == 'Foo')

bazArray = fooArray;

else

bazArray = barArray;

for (i=0; i < bazArray.length; i++)

baz.options[i] = bazArray[i];

}

</script>

</head>

<body>

<html:form name="MyForm" action="processMyForm">

<html:radio property="fooBar" value="Foo" onclick="changeOptions(this);"/> Foo<br/>

<html:radio property="fooBar" value="Bar" onclick="changeOptions(this);"/> Bar<br/>

Baz: <html:select property="baz"> </html:select>

</html:form>

</body>

</html>

**讨论**

您可以使用Struts来生成JavaScript，正如您可以使用它来生成HTML。一些开发人员认为JavaScript是“邪恶的”；实际上，它只是“有点邪恶”。采用一种实用的方法:如果JavaScript使应用程序更好，用户更满意，那么就使用它。但是，使用它时，您的业务逻辑会停留在业务层，而不是在web页面上。Struts帮助您做到这一点。

一个具体的例子可以说明这种方法。假设您想让用户选择他最喜欢的编程语言，然后为选择的语言选择最喜欢的集成开发环境(IDE)。该语言将使用单选按钮进行选择，并且IDE将从下拉菜单中选择。如果语言是Java，那么IDE下拉框将显示一些选项，如Eclipse、Net Beans、IDEA等等选项。如果语言是c#,那么下拉框将显示Visual Studio和SharpDevelop。

显示保存该数据的动作表单。

**示例3-9 ActionForm for favorite language/IDE**

package com.oreilly.strutsckbk;

This document was created by an unregistered ChmMagic, please go to http://www.bisenter.com to register it. Thanks

import org.apache.struts.action.ActionForm;

public final class MyForm extends ActionForm {

private static String[] javaIdes = new String[] {"Eclipse", "IDEA", "JBuilder", "JDeveloper", "NetBeans"};

private static String[] csharpIdes = new String[] {"SharpDevelop", "Visual Studio"};

public String[] getJavaIdes( ) {

return javaIdes;

}

public String[] getCsharpIdes( ) {

return csharpIdes;

}

public String getLanguage( ) {

return language;

}

public void setLanguage(String language) {

this.language = language;

}

public String getIde( ) {

return ide;

}

public void setIde(String ide) {

this.ide = ide;

}

private String language;

private String ide;

}

示例3-10显示了呈现输入页面的JSP（favorite\_language.jsp）。 此示例与解决方案类似。

**例3-10。 使用Struts渲染的DTHML的JSP页面**

<%@ page contentType="text/html;charset=UTF-8" language="java" %>

<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean" %>

<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html" %>

<%@ taglib uri="/WEB-INF/struts-logic.tld" prefix="logic" %>

<html>

<head>

<title>Struts - JavaScript Example</title>

<script language="JavaScript">

// Create the array for the first set of options

javaIdesArray = new Array( );

<logic:iterate id="ide" indexId="ctr" name="MyForm" property="javaIdes">

javaIdesArray[<bean:write name="ctr"/>] = new Option("<bean:write name='ide'/>", "<bean:write name='ide'/>", false, false);

</logic:iterate>

// Create the array for the second set of options

csharpIdesArray = new Array( );

<logic:iterate id="ide" indexId="ctr" name="MyForm" property="csharpIdes">

csharpIdesArray[<bean:write name="ctr"/>] = new Option("<bean:write name='ide'/>", "<bean:write name='ide'/>", false, false);

</logic:iterate>

function changeOptions(control) {

ideControl = document.MyForm.ide;

ideControl.options.length=0;

if (control.value == 'Java')

ideArray = javaIdesArray;

else

ideArray = csharpIdesArray;

for (i=0; i < ideArray.length; i++)

ideControl.options[i] = ideArray[i];

}

</script>

</head>

<body>

<html:form action="/admin/ViewFavoriteLanguage">

What's your favorite programming language?<br/>

<html:radio property="language" value="Java" onclick="changeOptions(this);"/> Java<br/>

<html:radio property="language" value="C-Sharp" onclick="changeOptions(this);"/> C-Sharp<br/>

<p>What's your favorite development tool?<br/>

IDE: <html:select property="ide"/>

</p>

<html:submit/>

</html:form>

</body>

</html>

在head元素中嵌套的脚本块包含JavaScript代码，logic:iterate标签循环遍历JavaBean属性，以创建两个JavaScript数组: 一个用于Java ide，另一个用于c# ide。每个数组都包含一组选项JavaScript对象。选项对象表示一个HTML选择控件的选项。该对象在构造函数中接受四个参数:显示的文本值，表单提交时传递的值，一个布尔值表示是否是默认选择的值，另一个布尔值表示当前是否被选中。

更改选项的JavaScript函数是在logic:iterate循环之后进行的。这个函数是纯静态JavaScript。触发更改的单选按钮将作为参数传递给函数。如果单选按钮控件的当前值是Java，那么选择控件将填充表示Java ide的Option对象。否则，控制将由代表c# ide的Option对象填充。

HTML主体包含表单，使用Struts HTML标记呈现。Struts标签通过onfunction属性支持JavaScript更改侦听器。对于单选按钮，onclick侦听器工作得很好。传递给函数的单个参数是对HTML单选按钮的引用。当页面最初呈现时，显示应该类似于图3-1。

**图3-1。 使用DHTML和Struts的表单**

一旦您单击了一个单选按钮，IDE字段下拉列表中的选项将填充来自表单bean的数据。图2-2在单击Java单选按钮时显示了显示。

**图3-2。 动态显示下拉菜单**

类似地，如果单击C-Sharp单选按钮，下拉列表中的值将更改为反映相应JavaScript数组的值。

可以使用JSTL代替Struts bean和logic标签。在本例中，您使用JSTL的c:forEach和c:out标签，而不是由未注册的ChmMagic创建文档，请访问http://www.bisenter.co进行注册。感谢logic:iterate和bean:write标签，这些标记以与Struts标记相同的方式生成JavaScript数组：

javaIdesArray = new Array( );

<c:forEach var="ide" varStatus="status"

items="${MyForm.javaIdes}">

javaIdesArray[<c:out value="${status.index}"/>] = new Option("<c:out value='${ide}'/>", "<c:out value='${ide}'/>", false, false);

</c:forEach>

JavaScript编程可能会让人沮丧，尤其是对于Java开发人员来说，它使用的是强类型和编译时检查。但是，提供这种类型的动态客户端交互可以带来更丰富的终端用户体验。

**另请参阅**

JavaScript：David Flanagan(o'reilly)的权威指南在JavaScript编程方面是无价的。如果需要业务逻辑来确定动态数据，那么目录3.9提供了更好的方法。