

# 目录

1. 关于本指南.....	3
2. 界面基础知识.....	3
2.1. ZK 基本知识.....	3
2.2. ZK 界面控件知识.....	3
2.3. ZK 界面构成原理.....	4
3. 界面布局设计.....	6
3.1. 总体布局方法.....	6
3.2. 确定元素顺序.....	6
3.3. 确定元素嵌套.....	7
3.4. 确定元素间距.....	8
3.4.1. 单个元素间距.....	8
3.4.2. 元素分组间距.....	11
4. 界面控件使用规范.....	11
4.1. 按钮.....	11
4.1.1. 按钮的位置.....	12
4.1.2. 按钮位置规则概览.....	13
4.2. 复选框.....	14
4.2.1. 排列对齐.....	14
4.2.2. 字段状态依赖于复选框.....	20
4.3. 单选框.....	22
4.4. 下拉列表和列表框.....	22
4.5. 表格.....	22
5. ZK 界面换肤样式.....	22
6. ZK 无障碍访问.....	23
7. ZK 界面模式.....	23
8. 附件：配套运行案例.....	23
9. 参考文档.....	24

# 1. 关于本指南

本指南描述了如何设计友好可用的 web 界面，并介绍如何基于 ZK 技术实现界面。内容主要包含以下方面：

- 介绍 Web 界面设计的知识和一般原则；
- 介绍总体布局规则和特定布局控件使用规范；
- 介绍每个 ZK 控件的外观、行为和界面属性；

本指南面向 web 应用开发者和 web 界面设计者。

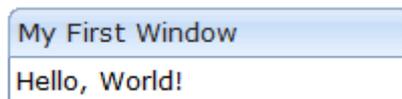
## 2. 界面基础知识

### 2.1. ZK 基本知识

ZK 是一套以 AJAX/XUL/Java 为基础的 web 应用开发框架，用于丰富 web 用户界面。ZK 最大的好处是让 web 应用开发像桌面开发一样简单。ZK 包含了一个以 AJAX 为基础、事件驱动、高互动性的引擎，同时还提供了丰富多样、可重复使用的 XUL 与 HTML 组件，及以 XML 为基础的界面设计语言 ZUML。

最简单的 ZK 界面代码如下：

```
<window title="My First Window" border="normal" width="200px">  
    Hello, World!  
</window>
```



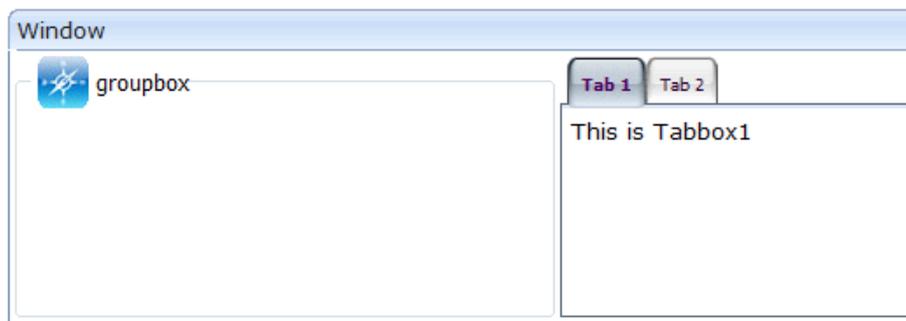
### 2.2. ZK 界面控件知识

Zk web 界面是由一系列的 zk 控件构成，所以必须了解这些控件才能更好的构建 web 界面。Zk 界面控件分为以下几类：

**基本控件：**基本控件是数据展示、数据输入、用户点击交互的一类控件，是用户界面的最基本构成，这类控件包含有：Label、Button、Input、Radio、Checkbox、Combobox、Bandbox、grid、tree、combobox、chart、slider、Splitter 等。

decimal box:	<input type="text"/>
date box:	<input type="text"/> Default <span style="border: 1px solid black; padding: 0 5px;">▼</span>
time box:	<input type="text"/>
positive int box:	<input type="text"/>
e-mail:	<input type="text"/>
multiple lines:	<div style="border: 1px solid black; padding: 5px; min-height: 40px;">text line1... text line2...</div>
Selection event:	<div style="border: 1px solid black; padding: 5px; min-height: 40px;">Welcome John !</div>
<input type="button" value="Replace"/>	

**容器控件：** 容器控件用于承载基本控件，用于对基本控件进行组织、分类、归纳。ZK中常见的容器类控件有：Window、Groupbox、Tabbox 等。



**布局控件：** 在容器控件容纳基本控件时，一般先要确定界面布局方式，这就需要布局控件进行界面布局的控件。布局控件一般为不可视，用于辅助确定基本控件的排列位置。Zk布局控件有：Borderlayout、Box、Grid 等。

### 2.3. ZK 界面构成原理

**示例单据**

名称:

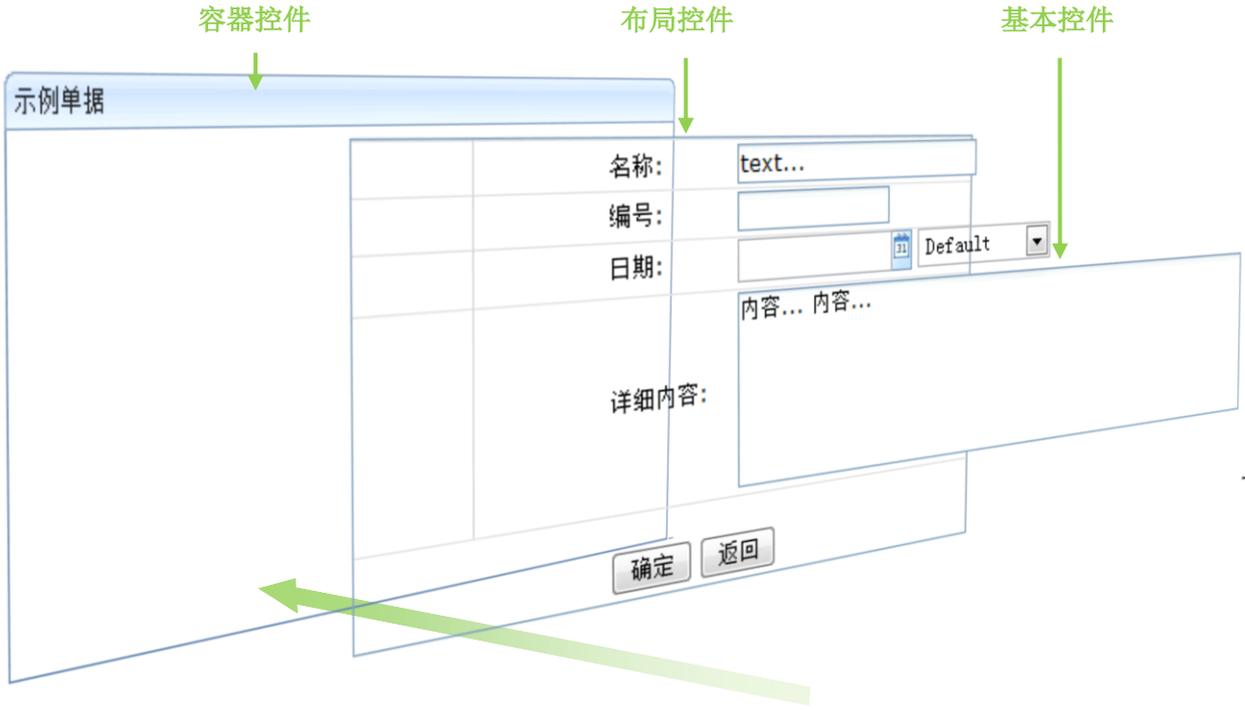
编号:

日期:  Default ▼

详细内容: 

内容... 内容...

界面效果图



界面构成图

界面代码如下:

```

<window title="示例单据" border="normal" width="490px">
  <grid width="480px" oddRowSclass="none" sclass="none">
    <columns>
      <column width="70px" /><column />
    </columns>
    <rows>
      <row>
        名称:<textbox value="text..." />
      </row>
      <row>
        编号:<intbox />
      </row>
      <row>
        日期:
        <hbox>
          <datebox id="db" />
          <listbox onSelect="db.setFormat(self.selectedItem.value)"
            mold="select" rows="1">
            <listitem label="Default" value="" />
            <listitem label="yyyy/MM/dd" value="yyyy/MM/dd" />
            <listitem label="MM-dd-yy" value="MM-dd-yy" />
          </listbox>
        </hbox>
      </row>
    </rows>
  </grid>
  <button label="确定" />
  <button label="返回" />
</window>

```

```
</row>
<row>
  详细内容:
  <textbox rows="5" cols="50">
    <attribute name="value">内容... 内容...</attribute>
  </textbox>
</row>
<row spans="2" height="45px" valign="bottom">
  <hbox>
    <button label="确定" /><button label="返回" />
  </hbox>
</row>
</rows>
</grid>
</window>
```

## 3. 界面布局设计

### 3.1. 总体布局方法

进行界面布局，并不是简单将界面元素“摆”在页面上，而要考虑以下方面：

- ◆ **操作流程**——用户如何在页面上完成他的工作；
- ◆ **关联关系**——页面上各个元素彼此间的相互影响；
- ◆ **归属关系**——哪些元素归属于哪些元素，元素之间关系的紧密、疏远；
- ◆ **美学和基本形式法则**——信息如何进行有效的视觉传达；

进行界面布局，可遵从以下步骤：

#### 1、确定元素的顺序；

依据操作流程、关联关系、归属关系，以先-后的线性形式确定界面元素的垂直、水平顺序关系。

#### 2、确定元素的嵌套；

依据关联关系、归属关系，以上-下层次形式确定界面元素的层级嵌套关系。

#### 3、确定元素的间距。

依据美学原则、关联关系确定不同层级上元素之间的间距。

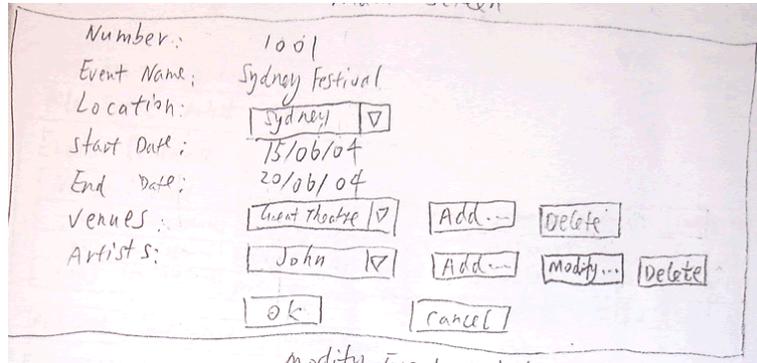
### 3.2. 确定元素顺序

界面元素的顺序一般是由用户在界面上的操作流程决定，界面设计师应寻找一种符合大多数用户习惯和任务场景的“自然顺序”。

例如，人的阅读习惯对界面元素的排列顺序有重要影响，所以界面元素的排列先是从左

到右，再是从上到下。另外，要遵从“要事优先”的法则，例如当填写个人信息的时候，一般是先填写姓名，然后才会填写其它的信息如性别、年龄、籍贯等。

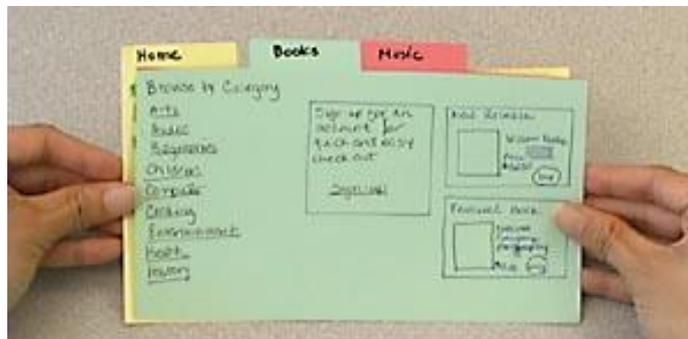
界面设计首先就是要确定界面元素的顺序，可以通过界面原型的形式表现，然后进行用户测试（如纸张原型，这时不用关注界面视觉细节）。



### 3.3. 确定元素嵌套

界面对象之间的复杂的层级关系、归属关系可以采用嵌套方式，嵌套是对界面信息进行分组、分类、分层次，可使用户更加高效的找到界面信息。但是嵌套意味着界面更加复杂，因为引入了更多的视觉元素：线条、边框、标题、间距、颜色等。

可通过纸张原型、HTML 原型等对界面嵌套进行调查，这时比第一步（确定元素的顺序）更注重一些界面的细节。



为了避免过于复杂的嵌套，应该遵循以下的嵌套规则：

**为避免多余的标题，禁止做以下事情：**

- 一个分组框在某个容器内，如 Window、Groupbox、Tabbox 内；
- 一个带标题头的表格在 Groupbox 内；

**为避免多余线框和杂边，禁止做以下事情：**

- 一个带标题头的表格在 Groupbox 内（这时表格标题头替代了 Groupbox）；
- 一个 Tabbox 在 Groupbox 内（这时用 Label 作为 Tabbox 的标题头）；
- Groupbox 内嵌套了 Groupbox（这时在 Groupbox 内用空行、分割线进行内分组）；
- Tabbox 内嵌套了 Tabbox，这要绝对避免，因为 Tabbox 的嵌套使得信息过度隐藏；

- 在容器控件之间放置 Separator;

详细界面嵌套规则如下表所示:

以下控件可以嵌套在右边容器内吗?	容器控件			
	Web 底层容器	Window	Groupbox	Tabbox
<b>Tabbox</b>	???	Yes	No*	No
<b>Groupbox</b>	Yes: 与其它元素一起; No: 单独一个	Yes: 与其它元素一起; No: 单独一个	No*	Yes: 与其它元素一起; No: 单独一个
<b>Listbox(Table), Tree</b>	???	Yes	No*	Yes
<b>Textbox, Image</b>	???	Yes	Yes: 与其它元素一起; No: 单独一个	Yes
<b>Splitter, Space</b>	???	Yes	Yes	Yes
<b>Label</b>	???	Yes: 用于标题	Yes: 用于标题	Yes: 用于标题
<b>Button, Combobox, Bandbox, Icon, Radio, Checkbox</b>	???	Yes	Yes	Yes

红色: 表示禁止嵌套; 黄色: 表示在特定环境下可嵌套; Yes: 表示可以嵌套; No\*: 表示常见的错误嵌套; ???: 表示应该考虑其位置。

### 3.4. 确定元素间距

#### 3.4.1. 单个元素间距

输入框之间垂直间距为 5px;

输入框之间垂直间距

姓名:	<input type="text"/>	5px
商业笔名:	<input type="text"/>	5px
E-mail地址:	<input type="text"/>	5px
确认E-mail地址:	<input type="text"/>	5px
密码:	<input type="text"/>	5px
确认密码:	<input type="text"/>	5px
生日:	<input type="text"/>	5px

Label 文本标签和输入元素之间水平间距为 8 ~ 22 px;

输入框之间垂直间距

姓名:	<input type="text"/>
商业笔名:	<input type="text"/>
E-mail地址:	<input type="text"/>
确认E-mail地址:	<input type="text"/>
密码:	<input type="text"/>
确认密码:	<input type="text"/>
生日:	<input type="text"/>

8~22px

输入框之间垂直间距

姓名:	<input type="text"/>
商业笔名:	<input type="text"/>
E-mail地址:	<input type="text"/>
确认E-mail地址:	<input type="text"/>
密码:	<input type="text"/>
确认密码:	<input type="text"/>
生日:	<input type="text"/>

间距过大



输入框之间垂直间距

姓名:

商业笔名:

E-mail地址:

确认E-mail地址:

密码:

确认密码:

生日:

间距过小



复选框、单选按钮之间垂直间距为 8px;

复选框之间垂直间距

复选框垂直间距

请选择下列中您感兴趣的话题,可多选.

外出郊游  游戏娱乐看电影  音乐

美食  海鲜  鲍鱼鱼翅

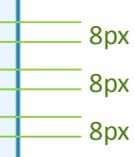
经济法律  IT行业  读书学习交流

艺术设计  工业品  管理

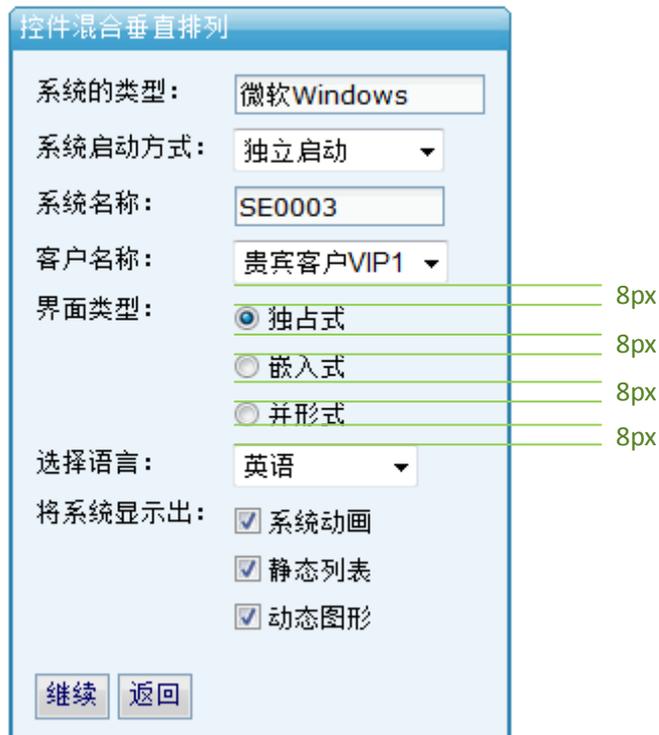
8px

8px

8px



多种元素混合垂直排列时,复选框和单选按钮边上的间距无论在什么情况下都为 8px,如下图:



### 3.4.2. 元素分组间距

- 窗口边框和内容区域的四周边距为 11px;
- 父组和子组之间的四周间距为 10px;
- 分组框边框和内部内容区域的四周边距为 5px;
- 复选框组、单选框组的组水平间距为 15px，如下图：



## 4. 界面控件使用规范

### 4.1. 按钮

按钮用于直接指向某个功能，或用于导航。

### 4.1.1.按钮的位置

将按钮放在所指向对象的底下，如果空间允许则将按钮放在所指向对象的右边。如果按钮指向多个对象，则将按钮放在最底部（最后面）对象的附近（底边或右边）。

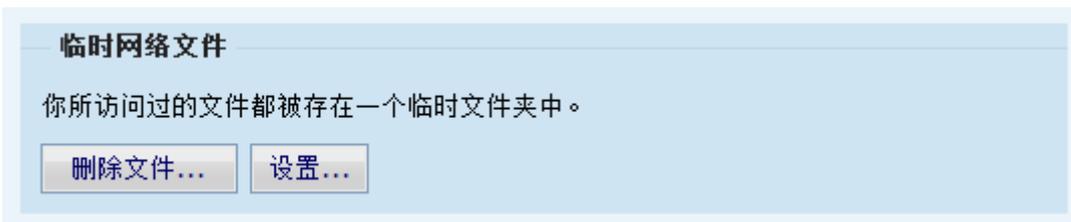
**如何清楚的表达按钮是指向哪个对象？**

- ◆ 将按钮放在对象内部，或放在对象附近，例如将按钮放在分组框内，并将按钮放在所指向输入域的旁边，如下图：



图：在一个分组框中的按钮指向一个下拉框

- ◆ 将按钮和所指向对象进行左对齐，如下图：

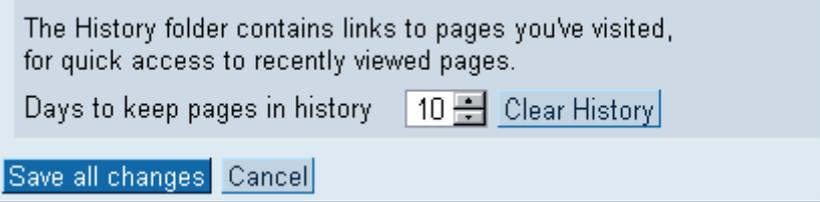


- ◆ 将按钮和所指向对象放在同一个层次上，如同一个分组框中、同一个区域中。
- ◆ 显示/隐藏对象：当对象隐藏时，对应按钮也要被隐藏。例如表格被隐藏时，所关联的按钮也要被隐藏。
- ◆ 最常用的按钮应该放在最左边。
- ◆ 导航作用（点击按钮后到达另一个屏幕）的按钮放在屏幕左下角位置，如下图：



#### 4.1.2. 按钮位置规则概览

以下表格对按钮摆放位置的规则做了总结。

指向对象	对象举例	按钮位置
单个对象	输入框	按钮仅靠在输入框的右边 
多个对象	界面元素组	按钮与最底部的对象左对齐，如果空间允许则仅靠在最底部对象的右边 
窗口区域，页签面板，分组框	分组框	按钮位于分组框左下角，并和分组框内其它元素左对齐 
表格，网格	表格	按钮位于表格下方，并和表格左对齐

Style	Value
Negative	-1.000.000,00
Positive	-3.000.000,00
Standard	0,00
SubTotal	2.000.000,12
<b>Total</b>	<b>2.000.000,35</b>

Back

## 4.2. 复选框

复选框为用户提供多选、单选、不选。复选框可组合成复选框组使用，复选框组以矩阵的形式排列，水平为 2~3 个复选框，垂直至多为 12 个复选框。复选框组适合于静态、相对少量选项的情况。对于动态、大量的选项，请参考使用“表视图”。

### 4.2.1. 排列对齐

#### 情况 1: 复选框与邻近的字段关联

复选框可能会与邻近的一个或多个输入字段关联。当复选框与一个输入字段关联时，将复选框放置在所关联输入字段的下方并左对齐，如图：



示例代码：

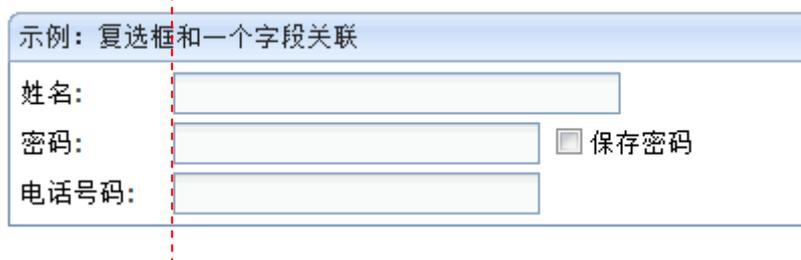
```
<window title="示例：复选框和一个字段关联" border="normal" width="320px">
  <grid width="310px" oddRowSclass="none" sclass="none">
    <columns>
      <column width="70px" /><column />
    </columns>
    <rows>
      <row>
        姓名:<textbox value="" width="220px"/>
      </row>
      <row>
        密码:<textbox type="password" width="180px"/>
        <checkbox checked="" type="checkbox" /> 保存密码
      </row>
      <row>
        电话号码:<textbox value="" width="220px"/>
      </row>
    </rows>
  </grid>
</window>
```

```

        </row>
        <row>
            <space/><checkbox id="spwd" label="保存密码"/>
        </row>
        <row>
            电话号码:<intbox width="180px"/>
        </row>
    </rows>
</grid>
</window>

```

这时如果空间允许，也可以将复选框紧靠放置在输入字段的右边，如图：



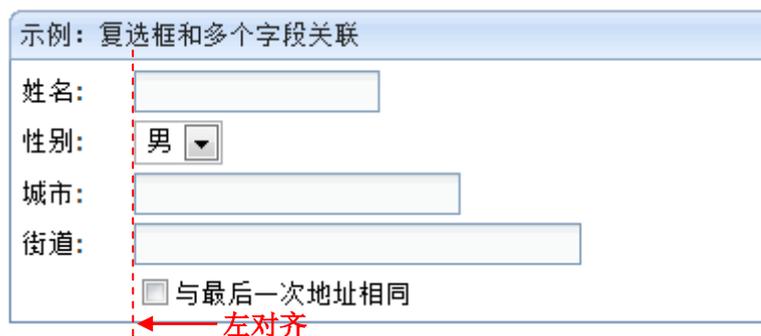
示例代码：

```

<row>
    密码:
    <hbox>
        <textbox type="password" width="180px"/>
        <checkbox id="spwd" label="保存密码"/>
    </hbox>
</row>

```

当复选框与多个输入字段关联时（与城市、街道同时关联），将复选框放置在最后面所关联字段下方，如图：



这时如果空间允许，也可以将复选框紧靠放置在最底部关联字段的右边，如图：

**示例：复选框和多个字段关联**

姓名:

性别:

城市:

街道:   与最后一次地址相同

## 情况 2：复选框与字段无关联，但在同一个组内

如果复选框在同一个组中，但是不存在与其它字段的关联关系，则将复选框的文本放置在前面，与其它文本标签左对齐，如图：

**示例：复选框和字段无关联**

姓名:

密码:

保存密码:

电话号码:

自动连接:

注意：这时为了界面排列整齐、容易阅读，将“保存密码”的复选框也同样放置。这时，技术上将复选框自身文本置空，使用 label 标签作为复选框文本。

示例代码：

```
<row>保存密码:<checkbox id="spwd"/></row>
<row>自动连接: <checkbox id="autosave"/></row>
```

另外，如果多个无关联关系的复选框形成组，那么可以在第一个复选框前面增加一个 label 标签作为组的标题，如图：

**示例：复选框和字段无关联**

姓名:

联系方式:  电话  
 邮件

城市:

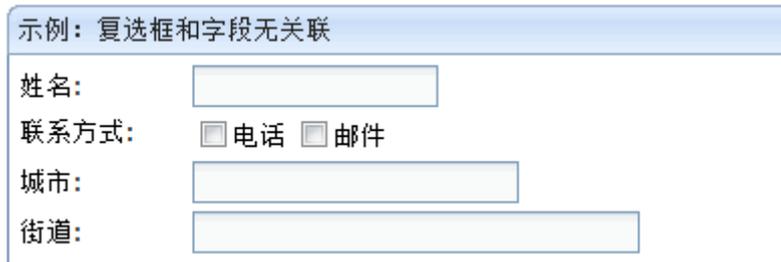
街道:

示例代码：

```
<row valign="top">
  联系方式:
  <vbox>
    <checkbox label="电话" />
    <checkbox label="邮件" />
```

```
</vbox>
</row>
```

这时如果空间允许，也可以使用水平排列的复选框组，如图：



示例代码：

```
<row>
  联系方式:
  <hbox>
    <checkbox label="电话" />
    <checkbox label="邮件" />
  </hbox>
</row>
```

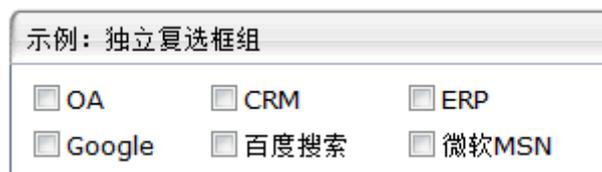
**错误举例：**



**错误原因：**因为这时的“自动连接”与上面的“密码”没有关联关系。只有存在关联关系时才使用复选框的这种布局。

### 情况 3：多个复选框排列成一个独立组

这时可以用分组框容纳这些复选框，复选框以矩阵的形势整齐排列，如图：



示例代码：

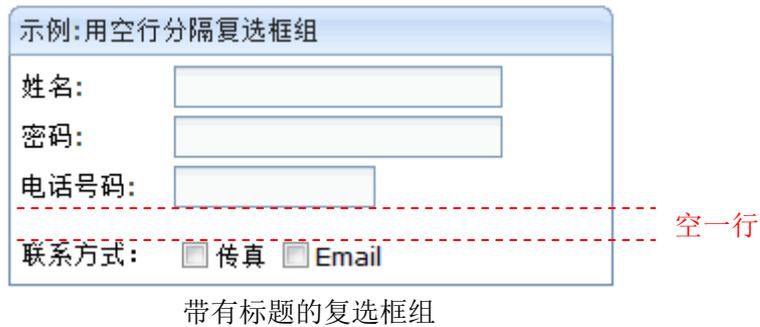
```
<groupbox mold="3d" width="300px">
  <caption label="示例：独立复选框组" />
  <grid width="100%" oddRowSclass="none" sclass="none">
```

```

<rows>
  <row>
    <checkbox label="OA" />
    <checkbox label="CRM" />
    <checkbox label="ERP" />
  </row>
  <row>
    <checkbox label="Google" />
    <checkbox label="百度搜索" />
    <checkbox label="微软MSN" />
  </row>
</rows>
</grid>
</groupbox>

```

也可以用空白的一行分隔形成独立复选框组。

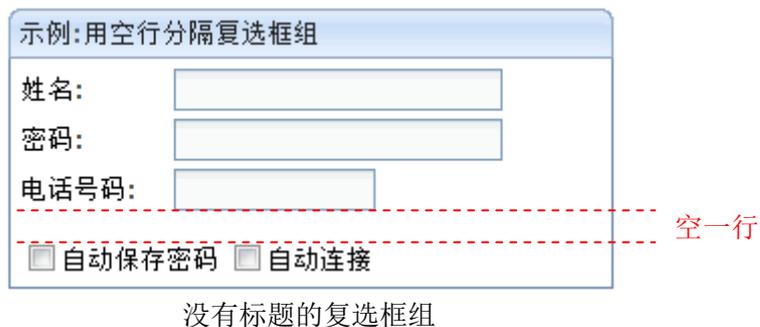


示例代码:

```

<row>
  <separator />
</row>
<row>
  联系方式:
  <hbox>
    <checkbox label="传真" />
    <checkbox label="Email" />
  </hbox>
</row>

```



示例代码:

```

<row>
  <separator />
</row>
<row spans="2">
  <hbox>
    <checkbox label="自动保存密码" />
    <checkbox label="自动连接" />
  </hbox>
</row>

```

这时如果复选框的数量比较多（大于 3 个），而横向空间有限，可将复选框垂直排列成组，如图：



示例代码：

```

<row>
  <separator />
</row>
<row valign="top">
  联系方式:
  <vbox>
    <checkbox label="传真" />
    <checkbox label="Email" />
    <checkbox label="电话座机" />
    <checkbox label="移动手机" />
    <checkbox label="普通邮件" />
  </vbox>
</row>

```

示例:用空行分隔复选框组

姓名:

密码:

电话号码:

---

自动保存密码

自动连接

限制10分钟后自动断开连接

示例代码:

```
<row>
  <separator />
</row>
<row spans="2">
  <vbox>
    <checkbox label="自动保存密码" />
    <checkbox label="自动连接" />
    <checkbox label="限制10分钟后自动断开连接" />
  </vbox>
</row>
```

### 4.2.2. 字段状态依赖于复选框

在某些情况下，输入框、下拉框等控件的状态取决于复选框是否选择。如下面的“联系方式”中，Email 是常用联系方式，默认复选框为勾选，Email 输入框为可输入状态。如图：

示例:字段状态依赖于复选框

姓名:

密码:

联系方式:  普通邮件

省份:

街道:

邮编:

Email

示例代码:

```
<window title="示例:字段状态依赖于复选框" border="normal" width="440px">
  <grid width="420px" oddRowSclass="none" sclass="none">
    <columns>
      <column width="70px" /><column />
```

```
</columns>
<rows>
  <row>姓名:<textbox /></row>
  <row>密码:<textbox type="password" /></row>
  <row valign="top">
    联系方式:
    <grid width="80%" oddRowSclass="none" sclass="none">
      <columns>
        <column width="50px" /><column />
      </columns>
      <rows>
        <row spans="2">
          <checkbox id="mail" label="普通邮件" checked="false">
            <attribute name="onCheck"><![CDATA[
              city.readonly=!self.checked;
              street.readonly=!self.checked;
              zipcode.readonly=!self.checked;]]></attribute>
          </checkbox>
        </row>
        <row>
          省份:
          <textbox id="city" width="130px" readonly="{!mail.checked}" />
        </row>
        <row>
          街道:
          <textbox id="street" width="180px" readonly="{!mail.checked}" />
        </row>
        <row>
          邮编:
          <textbox id="zipcode" width="90px" readonly="{!mail.checked}" />
        </row>
        <row spans="2">
          <checkbox label="Email" id="email" checked="true">
            <attribute name="onCheck"><![CDATA[
              emailtext.readonly=!self.checked;]]></attribute>
          </checkbox>
        </row>
        <row spans="2">
          <textbox id="emailtext" width="130px" readonly="{!email.checked}" />
        </row>
      </rows>
    </grid>
  </row>
</rows>
```

```
</grid>
</window>
```

### 4.3. 单选框

单选按钮必须 2 个以上成组使用。详细请参照复选框。

### 4.4. 下拉列表和列表框

暂无

### 4.5. 表格

暂无

## 5. ZK 界面换肤样式

在应用层引入样式表可更改 ZK 界面样式，代码如下：

```
<style src="/css/default/default.css" />
```

在样式表文件 `default.css` 中，应根据 ZK 官方发布的《ZK Style Guide》界面风格指南进行 ZK 界面样式的修改。如以下代码修改了按钮的样式：

```
button {
    background: #E7E7E7 url(/img/bgbtn.gif) repeat scroll 0%;
    border: 1px solid #7f9db9;
    padding-top:2px;
    font-family: Verdana, Tahoma, Arial, serif;
    font-size: small;
    font-weight: normal;
    color:#000066;
}
/* Not supported in <= IE6 */
button:hover { /* 鼠标悬停时按钮外观*/
    background: #E7E7E7 url(/img/bgbtnh.gif) repeat scroll 0%;
    border: 1px solid #7f9db9;
}
}
```



另外，定制 ZK 界面样式还可以通过以下方式进行：

- ◆ 对于某个特殊的组件，可以使用 `style` 和 `sclass` 属性去定制界面观感；
- ◆ 使用一个 `language addon` 去改变某种特定类型组件的所有实例的界面观感；
- ◆ 使用一个 `language addon` 去改变某中特定类型组件的所有实例的行为；

详细请参考 ZK 官方网站上的“Customizing Look and Feel, Part I - CSS”。

## 6. ZK 无障碍访问

美国联邦康复法案的第 508 节（简称 Section 508）于 1998 年由美国总统克林顿签署成为法律，这是界定对软件及其他电子和信息技术的无障碍环境要求。Section 508 是美国政府基于互联网协会的网页内容无障碍指导原则所制定的强制性法案，它规定了所有由联邦政府发展、取得、维持或使用的电子和信息技术都必须保证对残疾人群的无障碍。

[Section 508 中的一些部分 \(§ 1194.22 \)](#) 尤其与 ZK 相关。以下 ZK 无障碍条款是仅提供参考之用：

- A. 每个非文本元素，如图标、动画等，都应该提供对应的文本说明。在 ZK 中提供了 `setAlt()` 和 `setLongdesc()` 方法对 ZK 组件进行对应文本信息设置。

举例：

```
<image id="image" src="/img/sun.jpg" alt="Sunny day" />
```

- B. 对多媒体形式展现的，要提供同等意义的文本描述。在 ZK 中用 `toolbarbutton` 组件连接到同等的文本对象。

举例：

```
<flash alt="Creative Commons Flash Clip" src="img/cc.swf" />
```

```
<toolbarbutton label="transcript" href="http://www.cc.org/transcript.txt" />
```

- C. 用颜色表达的信息，同时也要提供文本的表达。在 ZK 中使用 `label` 组件表达颜色信息。
- D. 文档内容应该在没有 CSS 样式表文件时也能被良好阅读。这已经通过分离 CSS 样式在 ZK 组件中提供支持。
- E. 应该对 `iframe` 框架提供文本标题，以方便对框架进行识别和导航。在 ZK 中提供了 `iframe` 组件的 `setTitle()` 方法。

举例：

```
<iframe src="http://www.google.com" title="Search by google" />
```

- F. 页面上动画元素的闪烁频率应该在 2Hz ~ 55Hz 之间。Web 应用开发人员应该熟悉这些动画元素，如动画 GIF、Flash、Java-Applet 等，并遵循该标准。

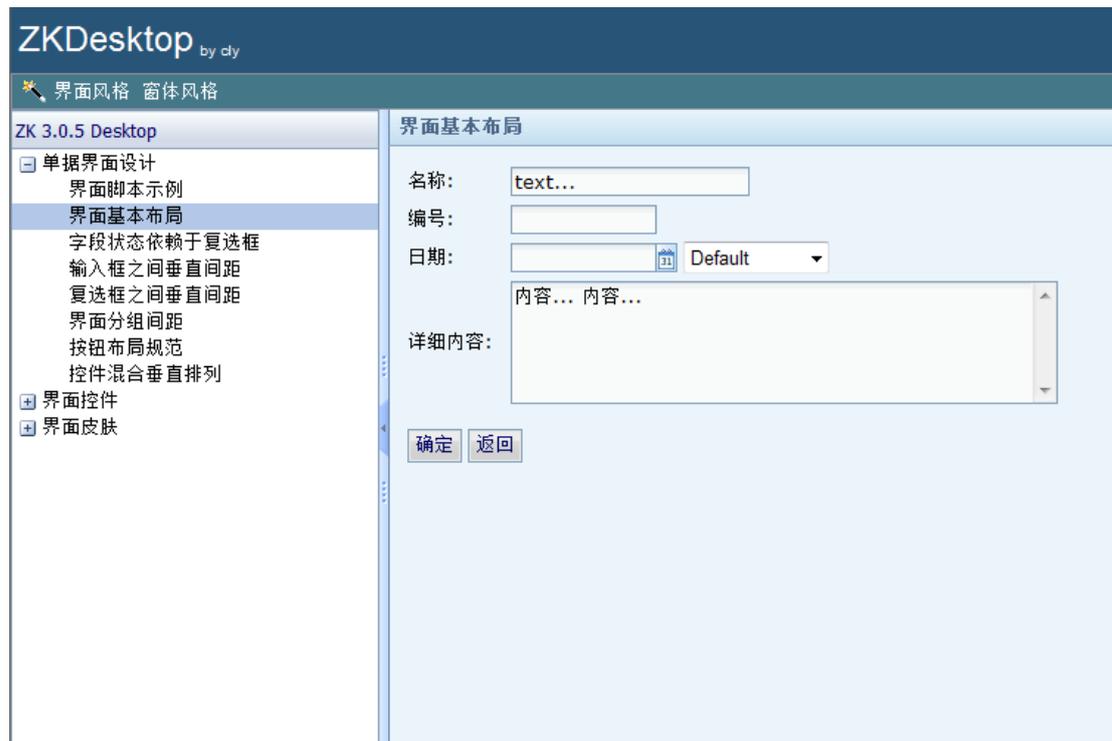
## 7. ZK 界面模式

暂无

## 8. 附件：配套运行案例

为了配合本文档，设计开发了 ZKDesktop，可以在 ZKDesktop 中直接查看本文档中所有

的界面布局方案、界面源码、换肤机制等。本文档的所有内容最终也将在这个 ZKDesktop 中体现，使得设计师、开发人员更加方便的阅读和查看界面设计和开发指南。



可在 Tomcat5.5 以上版本中直接运行 zkDesktop.war。

## 9. 参考文档

- 1、SAP Design Guide 上《HTMLB\_GuideLines》
- 2、ZK 官方上《ZK Style Guide》
- 3、ZK 官方上《Customizing Look and Feel, Part I - CSS》