

Web UI 规范

NSFT_IP02 V1.0

沈阳东软软件股份有限公司 版权所有
中国 沈阳浑南高新技术产业开发区 东软软件园
www.neusoft.com

更改履历

版本号	更改时间	更改的 图表和章节号	状 态	更改简要描述	更改申请 编号	更改人	批准人

注：状态可以为 N-新建、A-增加、M-更改、D-删除。

1.	目的	1
2.	主要部分	1
3.	目标读者	1
4.	基本概念	1
4.1	表单 (FORM)	1
4.2	控件 (CONTROL)	2
4.3	容器 (CONTAINERS)	2
4.4	事件 (EVENTS)	2
5.	控件功能	2
5.1	总述	2
5.2	语法规则	2
5.3	风格与样式	3
6.	浏览器平台	5
7.	常用界面风格	5
8.	表单	7
8.1	使用复选框 (CHECKBOX)	7
8.2	使用选项按钮 (OPTION BUTTONS)	11
8.3	使用不同的列表样式 (LIST BOXES)	16
9.	布局	26
9.1	总体布局策略	26
9.2	布局层次	28
9.3	组控件间的间距	32
9.4	单个控件间的间距	36
10.	布局控件	44
10.1	流程布局	44
10.2	表单布局	45
10.3	GRID布局	50
11.	可视化控件	53
11.1	导航栏	53
11.2	按钮	56
11.3	图表	58
11.4	复选框	65
11.5	日期导航	67
11.6	下拉列表框	69
11.7	文件上传	73
11.8	组	74
11.9	图像	78
11.10	输入域	82

11.11	链接.....	87
11.12	列表框.....	93
11.13	单选按钮.....	95
11.14	表视图.....	97
11.15	标签栏.....	106
11.16	文本编辑控件.....	111
11.17	树视图控件.....	114

1. 目的

本规范说明了如何开发可用的、可实现的网络应用程序的界面设计基本原则和指导。

2. 主要部分

用户界面设计的总体原则

用户界面的整体布局及控件布局的原则

用户界面中控件的参考实现，并说明每个控件的外观和行为

3. 目标读者

主要面向对象为网络应用程序开发人员，同时也可以给用户界面设计人员作为参考。如想更多的关注与本规范的最新进展，请访问东软开发者网站：<http://developer.neusoft.com>

4. 基本概念

以下介绍本规范中的一些重要基本概念。

4.1 表单 (Form)

表单基本上就是显示页面的封装，负责网络客户到网络浏览器的数据传输以及事件处理。表单中的控件必须有独自的控件名称。

4.2 控件（Control）

用于构建应用程序的图形化用户界面的基本要素。这些控件被放置在表单中。每个控件有其不同的用来定义控件外观的属性。控件有复选框，选项按钮和 grids 等等。

The figure displays several common web UI controls:

- Calendar:** A calendar for the year 2005, showing the month of March. The date 22 is highlighted.
- Table:** A table with 3 columns and 4 rows. The first column contains radio buttons, and the other two columns contain the text '内容' (Content).
- Form with Radio Buttons:** A form with a label '用户姓名*' (User Name*) and three radio buttons labeled '女士' (Ms.), '先生' (Mr.), and '公司' (Company). Below the label are input fields for '重复用户姓名*' (Repeat User Name*), '城市' (City), and '地址' (Address).
- Form with Checkboxes:** A form with a label '当前标题' (Current Title) and three tabs labeled '标题' (Title) and '长标题' (Long Title). Below the label are input fields for '用户名*' (Username*), '重复用户名*' (Repeat Username*), '城市' (City), and '地址' (Address). There are also checkboxes for '联系方式' (Contact Information) with options: '信件' (Mail), '电话' (Phone), '传真' (Fax), and '电子邮件' (Email).

图 4-1: 典型的控件

4.3 容器（Containers）

容器可以包含控件，也可以包括其它的容器。一个简单的容器就像一种能嵌套的“盒子”，其中包括表格状布局，在其中包含文本视图和输区域。

4.4 事件（Events）

组件可以对用户行为做出响应。这种响应就叫做事件。事件通常会引发提交（从网络客户提交表单数据到网络服务器）。通过这个能产生事件的控件，你可以明确事件处理的名称。Web 服务器接收到表单数据后，经过分析后，调用相关的事件处理过程从而进行下一步的处理。

5. 控件功能

5.1 总述

要使用控件，你必须了解这些控件的语法规则和属性。

5.2 语法规则

含有 JSP taglib 的程序遵循 XML 语句规则。每个控件都被包装（wrapped）在标签（tags）中，

通过 XML 前缀来明确的标识出标签（tags）。

5.3 风格与样式

门户软件必须能反映客户的企业标识和品牌规范，基于这个原因，需要提供了一种技术基础设施和工具来支持这个目标中的客户，需要提供了一定程度的设计灵活性，这种灵活性能够让我们的客户实现他们的品牌需要。

这种灵活性可以通过以下方式实现：

- 把所有设计信息放到级连嵌套表（CSS）中，而不用直接把它写入编码中。
- 在所有的控件中使用一个核心的 CSS
- 提供预定义的设计变量（如颜色模板）以便供用户选择
- 为支持客户网站的风格定制提供样式编辑功能

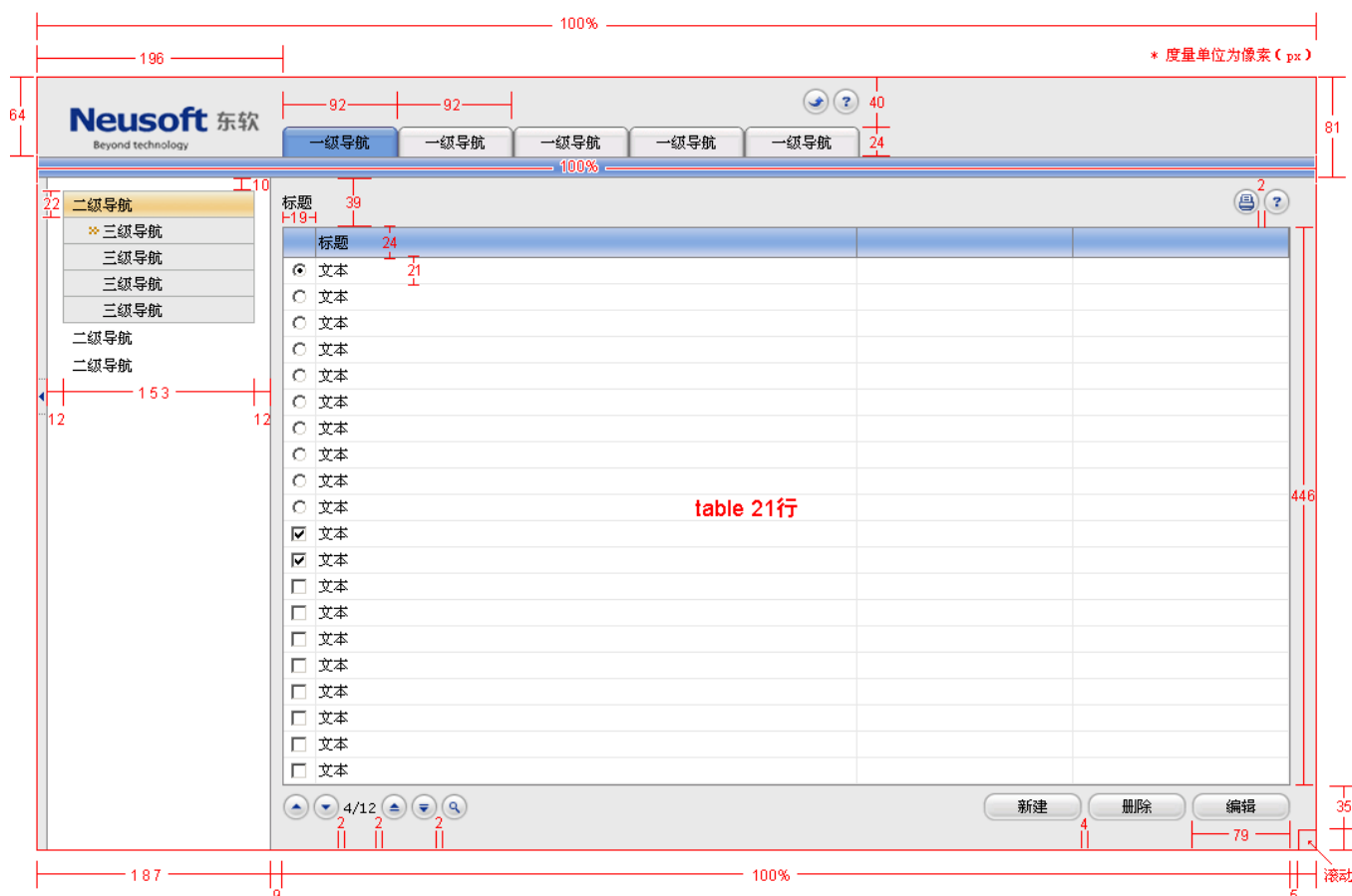


图 1：标准风格的门户表现

The screenshot displays a web application interface for Neusoft (东软). The header features the company logo and name, followed by a series of five '一级导航' (Level 1 Navigation) buttons. The left sidebar contains two menu items: '查看纳税人信息' (View Taxpayer Information) and '案件转出' (Case Transfer). The main content area is titled '标题 > 标题2 > 标题3' and contains a form with the following fields:

- 账号: (Account Number) - Text input field
- 名称: (Name) - Dropdown menu with '选项1' (Option 1) selected
- 方向: (Direction) - Radio buttons for '向上' (Up) and '向下' (Down), with '向下' selected
- 区分大小写: (Case Sensitive) - Checkbox, currently unchecked
- 描述: (Description) - Large text area with a vertical scrollbar
- 密码: (Password) - Text input field
- 部门: (Department) - Text input field
- 其他: (Other) - Text input field
- 时间: (Time) - Text input field
- 选择: (Select) - Dropdown menu with options '选项1' through '选项5' (Option 1 to Option 5)

At the bottom of the form, there are navigation controls including '4/12' and a search icon, and a row of four buttons: '新建' (New), '删除' (Delete), '保存' (Save), and '返回' (Return).

图2：用户定制的门户表现

6. 浏览器平台

在不同的浏览器平台上，哪些属性可编辑或不可编辑是有差别的。一般来说，选择Netscape4.7 比选择其它网络浏览器受限制。

对于目标系统采用Netscape4.7 的，以下的控件功能是不能被更改的：

- 指针：可点击和不可点击的指针
- 可输入域：可编辑和不可编辑的可输入区域的背景颜色。
- 按钮：按钮有默认的HTML外观（灰色和三维立体）；只有字体类型和大小可以通过常用界面风格修改。
- 文本编辑：使用默认的HTML要素，只有字体类型和大小可以通过常用界面风格修改。

7. 常用界面风格

所谓的常用界面风格，在内容显示区域中会影响到不止一个控件。为了避免每种控件多余的罗列，这里我们列出了一些常见的风格。

Internet Explorer 5 或以上系统的常用界面风格影响控件：指针、可输入区域、链接和文本。Netscape4.7 的常用界面风格只影响链接和文本。

Control	Style	IE 5 and above	Netscape 4.7
Cursor	Cursor for Clickable Elements	x	
	Cursor for Non-Clickable Elements	x	
Input Field	Background Color for Editable Fields	x	
	Background Color for Non-Editable Fields	x	
Link	Font Color for Unvisited Links	x	x
	Text Decoration for Unvisited Links	x	x
	Font Color for Active Links	x	
	Text Decoration for Active Links	x	
	Font Color for Links on Mouseover (Hover)	x	
	Text Decoration for Links on Mouseover	x	
	Font Color for Visited Links	x	x
	Text Decoration for Visited Links	x	x
Text			
Text Styles	Standard Font Family	x	x
Standard Text	Standard Font Size	x	x
	Standard Font Color	x	x
	Standard Font Style	x	x
	Standard Font Weight	x	x
Non-Standard Text	Font Size for Small Text	x	x
	Font Size for Large Text	x	x
	Font Size for Extra Large Text	x	x
	Font Style for Text Used as a Reference	x	x
	Font Color for Headlines	x	x
	Font Weight for Headlines	x	x
	Font Weight for Emphasized Text	x	x

表格 1 不同的浏览器平台和控件的常用界面风格

8. 表单

8.1 使用复选框 (Checkbox)

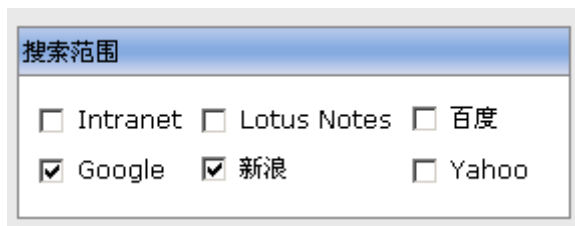


图 1: checkbox组

checkbox 为用户提供了一种或多种选择。用户可以什么也不选，或只选一种，也可以在一组中。

checkbox 要求多少种选择就选多少。

页面涉及到包含 checkbox 的排列，即：它们与输入域，表单种的其他元素之间的空间关系和从属关系。

checkbox组为用户提供了一整套的多种选择，这些选择或者水平排列（2-3 个checkbox）或者垂直排列（不超过约 12 个checkbox），又或者是按类似矩阵的样式排列。

排列

关于 checkbox 的位置排列我们可以区分为以下几种案例：

- 关于邻近区域的 checkbox
- 与元素无关但应该包括在区域组中的 checkbox
- 可以列为独立信息区的 checkbox

用例 1：与一个以上区域相关的复选框

把从属的复选框和其它输入元素的左边界对齐（如图 8-2 所示）。把checkbox标签放在checkbox的右侧（对于checkbox控件功能，这是自动形成的）

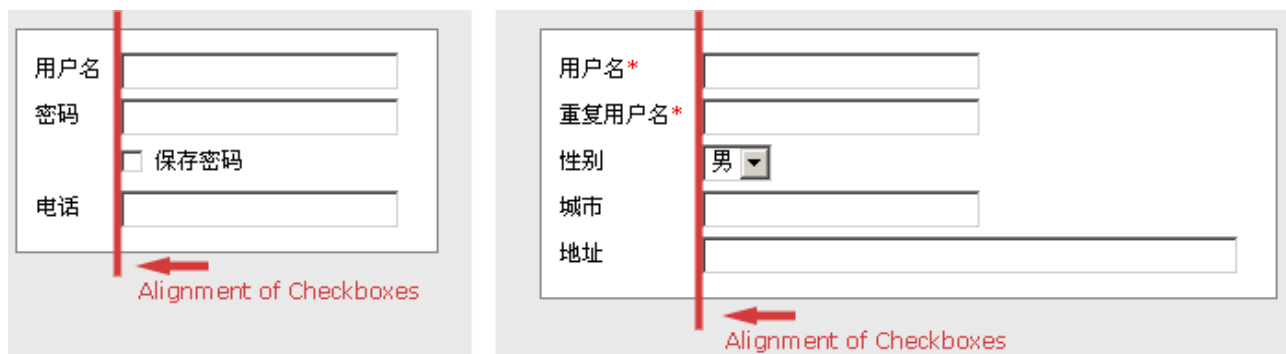


图 1a-b: 与其上方的一个可输入区域相关的checkbox（左图），或与两个可输入区域（城市、

街道) 相关的checkbox (右图)

除了上述之外, 还有另一种排列方式。如果空间允许的话, 可以将一个单独的checkbox放在参考区域的右侧。如果有不止一个参考区域, 将checkbox放在最下面的参考区域的右侧。

图 1c: 与其左侧的可输入区域相关的checkbox(等同于图 1a)

用例 2: 包含在区域组中的 checkbox

如果复选框包含在区域组中, 但却与特定的区域无关。则将checkbox 标签放在左侧, 并且使复选框与其它可输入区域对齐。(如图 2a 所示)

注意: 在这种情况下, 你必须使 checkbox 文本靠近一个空白文本并且使用针对标签的标签控件功能。

除了上述之外, 还有另一种排列方式: 你可以增加一个与这个组的其它标签左对齐的标签, 并且使用checkbox文本提供额外的信息(如图 2b所示)。那样的话, 只有第一个checkbox有说明整个组的标签。

图2a-b: 在一个区域组中左边有标签或有两个标签的checkbox

如果空间允许, 你也可以使用占用一行的水平排列的 checkbox 组 (比较典型的是 2—3 个复选框, 如图 2c 所示)

Figure 2c-d shows two examples of checkbox groups. The left example (Figure 2c) shows a form with labels and checkboxes for contact information. The right example (Figure 2d) shows a form with labels and checkboxes for user preferences.

Figure 2c (Left):

- 用户名* [text input]
- 重复用户名* [text input]
- 联系方式 ☐ 邮件 ☐ 传真
- 城市 [text input]
- 地址 [text input]

Figure 2d (Right):

- 用户名 [text input]
- 密码 [text input]
- 电话 [text input]
- 自动通知 ☐

图2c-d: 一个区域组中水平排列的checkbox组（左图），没有组标签的缩进的checkbox（右图）

注意：在这种情况下（图 2d），不要使用没有组标签的排列，因为可能会引起歧义。这样的布局表明了从这个组上方区域到用户的从属关系。尽管图 2d 中的布局与图 1a 相同，图 2d 的使用仍是错误的，因为 checkbox 与密码区无关。因此只有在存在从属关系的情况下，才能用这种布局。（例 1）

用例 3：构成独立信息区域的复选框

如果复选框排列在一个 checkbox 组中，它们与其它标签左对齐，并且以类似矩形的样式排列，这样的组或者包含在一个组控件中（如图 3a 所示），或者用空白与区域组隔开（如图 3b 所示）

Figure 3a-b shows two examples of checkbox groups. The left example (Figure 3a) shows a group of checkboxes for search range. The right example (Figure 3b) shows a group of checkboxes for user preferences.

Figure 3a (Left):

- 搜索范围
- ☐ Intranet ☐ Lotus Notes ☐ 百度
- ☒ Google ☒ 新浪 ☐ Yahoo

Figure 3b (Right):

- 用户名 [text input]
- 密码 [text input]
- 电话 [text input]
- ☐ 保存密码 ☐ 自动通知

图 3a-b: 构成自己的信息单元的checkbox组——或者包含在一个组中（左图）或被空行隔开（右图）

如果仅有几个复选框，你可以不用矩阵，而使用水平排列方式。

水平的 checkbox 组有两种可能的排列：

在 checkbox 左侧可以加入标签以便介绍 checkbox 的特性，在这种情况下，把复选框和其它要素对齐并且使用针对标签的标签控件功能。（图 4a）

checkbox 行没有介绍的标签（图 4b）

用空行把水平 checkbox 组和前面的区域隔开。

注意：水平checkbox 组的一种极端案例就是组种仅有单个的checkbox.。

图 4a-b: 水平的checkbox组的示例，或者在其左侧有介绍的标签（左图），或者没有介绍的标签（右图）

对于 2---3 个以上的复选框，有或没有标签的垂直排列方式更适合。（见图 5a和 5b）

图 5a-b: 垂直的checkbox组的示例，或者在其左侧有介绍的标签（左图），或者没有（右图）

从属区域

在某些用例下，可输入区域、下拉菜单、或其它的控件，它们的状态可能取决于checkbox的设置，下面我们举一个简单的示例，在这个示例中，用户输入自己的联系方式的信息。一些不能被选中的checkbox显示的是默认值，它下面的输入区域是只读的。能被选中的checkbox代表的是一些特殊的情况。用户选中checkbox，输入的部分就可以填写相应信息了。

图6：控件其下方可输入区域的可编辑性的复选框

如果有更多的从属要素，将从属的组缩进使标签与其它可输入区域左对齐（最上面的 checkbox）。如果仅有一个从属区域，通常不需要区域标签（最下面的 checkbox）。

相关控件

Radio Button, Dropdown List Box, List Box, Label, Grid Layout

8.2 使用选项按钮（Option buttons）

选项按钮可以让用户在一组选项中作出单一的选择

图1：radio button组

这个页面包含了选项按钮的排列，即它们与区域，与类似格式的结构中其它要素之间的空间关系和从属关系。

Radio button 组为用户提供了一组选项，这些选择或者水平排列（2—3 个 radion buttons）或者垂直排列（不超过约 12 个选项按钮），又或者按类似矩阵的样式排列。

排列

关于选项按钮的位置排列，我们可以区分为以下几种情况：

- 关于邻近区域的选项按钮
- 与元素无关但应该包括在区域组中的选项按钮

- 可以列为独立信息区域的选项按钮

一般来说，前两种情况在 checkbox 中更为常见：

案例 1：与一个或一个以上区域相关的选项按钮

使相关联的选项按钮与其它输入要素的左边界处对齐（图 1）。把 radio button 标签放在 radio button 的右侧（对于 radio button 控件功能，这是自动形成的）

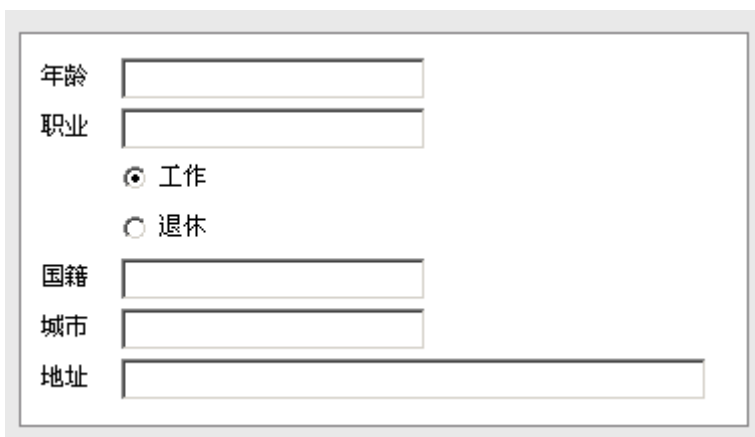


Figure 1 shows a form layout with the following elements:

- 年龄 (Age): Input field
- 职业 (Occupation): Input field
- 工作 (Work): Radio button option
- 退休 (Retire): Radio button option
- 国籍 (Nationality): Input field
- 城市 (City): Input field
- 地址 (Address): Input field

The radio buttons are aligned with the left edge of the input fields.

图1 与其上方的可输入区域(职业)相关的一对选项按钮

注意：如果有太多的选项，考虑一下使用下拉菜单而不用 radio button 组。

如下所示，如果与 radio button 组上方的其它区域有从属关系，才使用这种排列。

用例 2：包含在区域组中的选项按钮

如果选项按钮包含在区域组中，但却与特定的区域无关。则按如下操作（图 2a-b）：

- 把 radio button 组的说明放在组的左侧，并且使它与其它区域标签对齐。
- 把选项按钮与其它可输入区域对齐。
- 使用组标签的标签控件功能

你可以垂直排列 radio button 组，或者如果空间允许的话，可以水平排列只占一行的 radio button 组。

图 2a-b: 在区域组中左侧带有组标签的选项按钮（左图是垂直排列，右图是水平排列）
总而言之，区域组中的radio button组应该有一个说明性标签，并且每个radio button的右侧有一个各自标签。

基本原理：一个 radio button 组在功能上等同于一个下拉菜单。组标签相当于下拉菜单的标签，选项按钮右侧的标签相当于菜单项。

图2c-d: 区域组中在其左侧有两个标签的radio buttons(左图), 右侧有两个标签的选项按钮(右图)

切忌

Radio button组没有标签的布局或标签在左侧的布局是不易于理解的，因为标签不是像包含的区域标签那样在同一个语义层次。

在例 2（图 2d）中也不要使用没有组标签的排列，因为这样也很难懂而且可能会引起歧义，这样的布局表明了从这个组上方区域到用户的从属关系。尽管图 2d 中的布局与图 1a 相同，图 2d 的使用仍是错误的，因为选项按钮与“Last name”区域无关。因此只有在存在从属关系的情况下才能用这种布局（例 1）

用例 3：构成独立信息区域的选项按钮

如果选项按钮排列在一个单独的radio button组中，那么把选项按钮以类似矩阵的样式排列，与页面或应用程序中的其它要素左对齐。这样的组或者包含在一个组控件中（如图 3a所示），或者用空白与区域组隔开（如图 3b所示）

图 3a-b: 构成自己的信息单元的radio button组—或者包含在一个组中（左图）或被空行隔开（右图）

如果仅有几个选项按钮，你可以不用矩阵，而使用水平排列方式。在这种情况下，把radio button组的列数这个属性设置一个值，使整个radio button组只有一排。

水平radio button组有两种可能的排列：

radio button行的左侧可能有标签或其上方有标题，它们是用来介绍radio button行的，在这种情况下，把选项按钮和其它要素对齐并且使用针对标签或标题的标签控件功能。（图 4a-b）radio button行没有介绍的标签（图 4c-d）

用空行把水平radio button组和前面的区域隔开。

图 4a-d: 水平radio button组，或者左侧有介绍的标签（左上图）标题（左下图），或者没有介绍的标签（右图）

注意：不要使用单一的 radio button，因为会有两个问题：（1）用户无法对单一的 radio button 取消选定；选定它后，仍然无法取消。（2）用户只能看见一个选择的名称，至于另一个选择是什么，他们通常只能猜测。

从属区域

在某些情况下，可输入域、下拉菜单、或其它的控件，它们的状态可能取决于radio button组的设置，下面我们举两个简单的示例，在这两个示例中，用户输入自己的国籍或支付方式（图 5b）。第一个radio button显示的是默认值，它下面的输入区域是只读的。第二个radio button代表的是一些特殊的情况。如果用户设置了radio button，从属输入区域就可以填写相应信息了。除此之外，还有一种方法：在第一个示例中，如果知道选项的内容并且选项的数量不是很多，可以用下拉菜单代替可输入区域。

The figure consists of two side-by-side form examples. The left example shows a vertical list of labels: '用户名 *', '重复用户名 *', '国籍', '城市', and '地址'. '国籍' has two radio buttons, '中国' and '其他'. Below the radio buttons, there are input fields for '城市' and '地址'. The right example shows a similar layout but with more options. '支付方式' has two radio buttons, '现金' and '信用卡'. Below '信用卡', there is a dropdown menu for '卡' (currently showing '建设银行龙卡'), followed by input fields for '卡号' and 'Expired'.

图 5a-b: 控件这个组下方一个（左图）或者几个（右图）可输入区域的可编辑性的垂直radio button组

垂直排列的 radio button 组控制一个可编辑输入域（左侧）和多个可编辑输入域（右侧）

如果从属要素不止一个，不要使用这种布局。在这种情况下，应该缩进从属组使标签与其它可输入区域左对齐。（如图 5b 所示）

设计选择

选项按钮与下拉菜单还有与为用户提供单一选择相关的下拉菜单在功能上是相似的。如果选项极少(2—6 个)，并且用户要迅速看到所有选项的情况下，可以使用选项按钮。

相关控件

Dropdown List Box, Checkbox, List Box, Label, Grid Layout

8.3 使用不同的列表样式（List Boxes）

列表的设计选择

本规范中提到的控件功能涉及到显示、编辑数据设置以及从数据设置中进行选择。

- **Checkbox:** 从少量数据集中作多项选择（静态）
- **Radio Button:** 从少量数据集中作单一选择（静态）
- **List Box:** 从少量数据集中作单一选择（静态）
- **List Box:** 从少量到中等规模的数据集中的单一选择（动态或静态）
- **Item List:** 在一列中显示少量到中等规模的数据集—可以是有顺序的或无序的（静态）
- **Table View:** 在一个或更多列中显示和编辑各种显示变量的数据集（动态），可能是单一选择、多项选择。

想了解它们各自的控件功能信息，参见Checkbox, Radio Button, List Box, Item List, 和 Table View.

下面我们要详细地讨论设计选择，特别是关于重叠的应用区域方面。

注意：这里没有讨论 checkbox 和 radio button groups 的用例，想详细了解这些控件功能的布局选择，参见表单 – 使用 复选框 和 表单 – 使用 选项按钮

下拉列表与列表的对比

以上的总结表明了列表与下拉列表在功能方面是相似的，它们都提供了一系列选项，用户可以从选取一项，也就是说，它们都是单选列表。在这你可以找到在这两种控件功能中作选择的标准，除此之外，我们还为一系列选项按钮是否适用提供了相关提示。

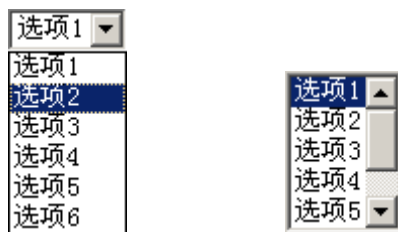


图1：下拉列表（左图）与列表的对比

什么时候使用列表？

- 如果页面上有更多空间时

- 如果选项数量多
- 如果用户需要了解当前选择的应用场景，即元素集或至少是元素集的一部分
- 如果用户需要仔细考虑他们的选择
- 如果用户的鼠标的性能不太好

什么时候使用下拉列表？

- 在表格视图中
- 如果空间有限—只能占一行
- 选项数量较少（不超过 20 项）
- 如果用户只需要了解当前的选择，而不用了解整个集合

什么时候使用 radio button？

选项极少(2—6 个)，并且用户要迅速看到所有选项的情况下，可以使用选项按钮。

更大的radio button集的使用仅限于一些特殊情况中，例如：所有选择必须是可视的，用户对这方面的专业知识不是很在行的，并且/或应用模拟一种书面的格式，比如基于网络的调查问卷或定单表格。

注意：对于多项选择，应使用复选框来取代选项按钮

元素列表与列表框的对比

元素列表和列表框都可以用于显示选择项，而列表框还可以在显示的一系列的选项中作选择。元素列表是静态的列表，外观类似于纸面；它们可以是有序的也可以是无序的，但列表框外观类似于表单，而且列表框不止包含可以看到的项目。

如果你想使用列表框，你可以选定表格是否也可以作为一种可利用的设计选择。我们也为这种选择提供了一些提示。

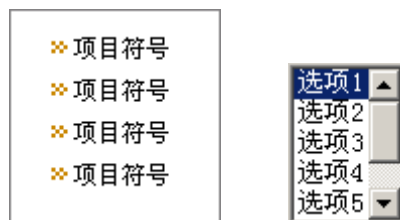


图2：元素列表（有序的, 左图）与列表框的对比

什么时候使用元素列表？

- 类似于纸面的应用，比如新闻、文章等

- 如果可能显示所有项目或者作为一个可滚动的页面或应用。
- 如果项目的数量是固定的

什么时候使用列表框？

- 在表单类应用中，特别是带有其它表单要素，比如可输入域和选择要素
- 如果需要作选择
- 如果空间不够显示整个列表
- 如果页面或应用不能滚动显示更多元素
- 如果不应该显示数字

什么时候使用表格视图？

下面，我们列举了几个应用场景，在这些场景中表格视图可以用来代替列表框。注意，我们并不推荐使用表格视图，它只是一个可选项，唯一的例外是多项选择，它只能使用表格视图。

具有代表性的是，在透明化的设计中你会用到表格视图。注意表格视图会衍生出“visual overhead”风格，比如标题，首列，滚动按钮

几种使用表格视图的情况：

- 在表单风格的应用中，来自于不同的信息单元无法组合到区域中。
- 需要多列显示时
- 滚动按钮优于滚动条时（页面方式滚动）
- 需要多项选择时（通常用列表框是无法实现的）

注意：静态的多项选择也可以通过使用 checkbox group 来实现。

相关控件

Checkbox, Item List, Dropdown List Box, List Box, Radio Button, Table View

表格视图功能

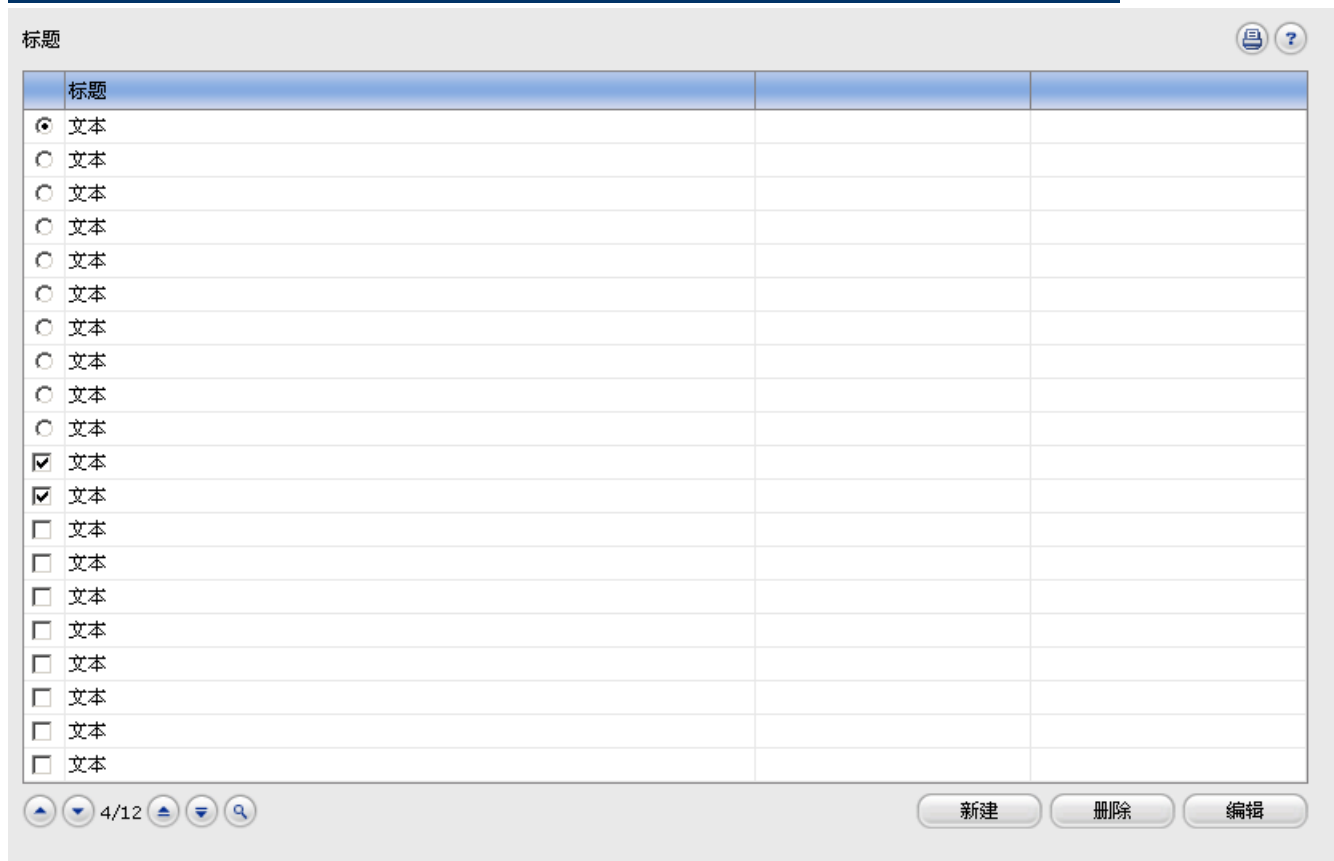


图1：带有不同列类型的表格视图和错误输入区的示例

表格视图不仅以列表的样式显示数据、允许表格数据的输入，而且提供了许多功能，其中一部分功能可以改变数据显示，其它功能可以改变数据本身。

表格视图功能与作为一个整体的表格视图相关，也与列、行（等同于元素）、或一个单元格。后面会逐一论述这些示例，除此之外的常用的如：排序与过滤等功能也包括在内。

注意：这里我们没有包括包含在表格视图框架中的功能，比如滚动功能。想了解这些功能，参见 Table View

想了解表格视图控件功能的详细信息，参见 Table View

功能按钮的位置排列

位置

把表格视图中的按钮放在表格视图的下面并且与表格左对齐。如果有重要的按钮把它放在左侧。

按钮分组

用分隔符把按钮组隔开（无空格）

与表格视图列、排序相关的功能

与列相关的按钮

例如排序的一些功能与特定的列相关，把带有小的图标的按钮放在首列中来提高交互响应的速度。

如果一个功能中有不止一个特性（例如，不同排序或统计），为了节省空间，考虑一下一次只显示一个图标，不要同时平行显示两个或更多的图标。

在一个列头中使用的图标的数量不要超过三个。

列头中的链接

如果功能要求如此，你也可以在列头文本中使链接。

与表格视图行相关的功能

与表格行相关的功能通常与显示在一行中的元素、数据相关。这些功能可以表示为：

按钮

图标

链接

大多功能使用按钮，只有在下面的特例中才使用图标和链接：

- 图标：网络标准的，比如删除、信息、帮助或购物篮
- 链接：导航功能，比如详情（把链接放在名称或ID列中）、更多信息等

按照选择正确控件功能的一般原则，要考虑应用环境及各自的“网络标准”。

表格单元：链接与按钮的对比

注意：链接本身没有什么自身的解释说明，因此使用链接时必须清楚地了解链接的目的，给链接加上提示支持用户使用。

过滤

尽可能多地提供过滤功能，过滤功能帮助减少表格视图中显示的数据的数量。这会帮助用户改善性能。

布置

把 shuffler 声明放在表格视图的上方并且与它左对齐。

原因：如果窗口大小改变了，左对齐的shuffler不会隐藏于视图中。

按钮

使用标有“Go”的按钮开始过滤过程，只有在简单过滤的情况下使用下拉菜单中的 events（带有标签的下拉菜单）

相关控件

Tree View, Item List, List Box 树状视图, 项目列表, 菜单

定位按钮

按钮用于目的明确的功能, 这些功能与特定的目的相关或为导航目的服务的。

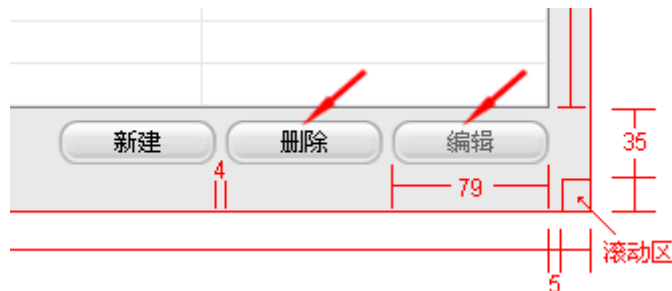


图 1: 示例: 包含带按钮的组的视图和属于视图本身的两个按钮。

定位

把按钮放在它们相关的目标下方。如果空间有限, 把按钮放在目标的右侧 (如果有多个目标, 那么把按钮放在位于最下方的目标的右侧)

如何确定一个按钮与哪个目标相关:

把按钮放在目标内, 或靠近目标。

示例: 把按钮放在group boxes内, 或把按钮放在与它们相关的区域附近

图 2: 按钮在与列表框相关的组内的示例

右排列按钮与对象

示例: 把和区域相关的按钮与区域标签右对齐。

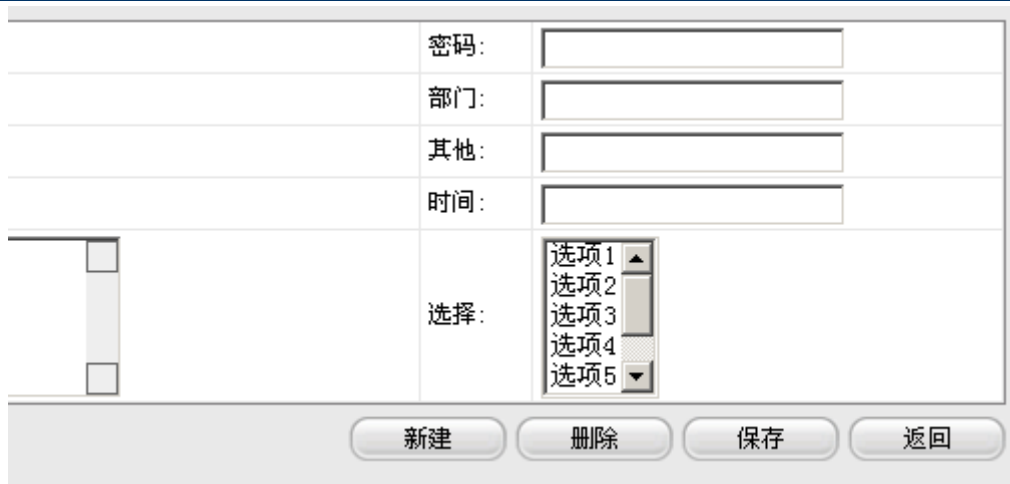


图 3：组内右对齐的按钮的示例

把按钮放在与对象相同的层次上（显示区、分组框），这些对象受行为影响

示例：把与区域相关的按钮放在分组框里，或分组框里的区域，或区域中（图 2 和图 3）

显示/隐藏目标：当目标隐藏时，按钮也隐藏

示例：如果表格隐藏，相关按钮也隐藏



图 4：示例：包含重要按钮的左对齐按钮组

如果重要按钮是按钮组的一个组成部分，那么它应是这个组最左侧的按钮

导航按钮放在屏幕（或屏幕区域）的左下方

定位规则的总结

下面的表格总结了按钮放置的规则

Web UI 规范		
对象	示例	定位
单一对象	区域	对象右侧
多个对象	区域组	
区域、分栏、分组框	分组框	
表视图	基于表视图控件的表	
特殊用例：长表格	滚动表格	图6：表格下方的按钮在表格的上方和下方，左对齐。替代方案：为按钮在表单上方实现特定的不包括滚动特性的帧

表格 1：按钮位置放置的规则

不同的按钮类型和大小的使用规范在Button一节有论述。

相关控件

Link, Input Field, Group, Table View

处理错误

除了帮助，处理错误也是用户支持的一个重要方面。错误处理可以帮助用户应对有问题的情况，继续他们的工作。

通常，处理错误是通过显示错误发生的位置和发送信息来进行的，这个信息记录了错误，说明了错误原因，更理想的信息还可以提供关于如何改正错误的提示。

这页包括三部分：（1）错误阻止 （2）处理区域中的错误 （3）处理表格中的错误

错误阻止

最首要的是阻止错误！

在处理错误前，你首先应该考虑如何防止错误的发生。通常，你应该设计网络应用程序使错误不会发生。阻止错误而不是改正错误有以下好处：

- 用户不能陷入错误的困境中—许多用户都不太容易从错误中走出。
- 用户的工作不会被错误信息打断。
- 用户不会被错误信息（通常是隐藏的）所困扰。
- 屏幕区域不需要显示错误。

如果你阻止不了错误，请遵循以下指导：

如何阻止错误

通常需要重新考虑并且放弃“旧习惯”来寻找设计解决方案，这个

解决方案是用来防止错误的，而不是在错误发生之后发送错误信息。

下面我们提供了一些办法和示例，或许会为你寻找如何阻止错误的方法带来些灵感。

阻止错误或无效的输入—总论

使用准确的表述和说明，不要太简化（特别是在网络应用程序中）

说明必输入的区域（用一个红色的星号 * ）和一个说明的文本。

阻止错误或无效的输入

- 数字域：通过解析输入串，阻止用户输入字符
- 日期和时间域：提供预格式化的“智能”日期和时间域，或者提供选择控件功能来替代可输入区域（下拉列表，按钮，日历控件）
- 金融相关输入域：使用预格式化的域

防止不完整的输入

说明要求的区域（用一个红色的星号 * ）和一个说明的文本

防止无效操作

- 将不能在当前环境下使用的按钮设为未激活状态
- 不提供不需要的功能

阻止灾难性操作

- 如果某些操作会给用户带来严重的后果，给相关的按钮加上说明的文字，告知用户可能出现的后果。
- 如果用户可能会丢失数据，显示对话框。

以正确的和有目的性的方式使用控件

如果有从属关系或必须遵循特定顺序的步骤，那么不要使用用户期待以任何次序来应用的屏幕要素。

示例：不要在视图中使用彼此影响 Tab 栏，至少不要强迫用户按固定的顺序执行某些步骤。

总的来说，不要以无目的性的方式使用控件功能。“创造性”的控件功能的使用与用户的期待不相符，并可能引起严重的使用问题。

示例：不要仅仅因为你更喜欢复选框的外观而错误地用复选框来代替选项按钮

使用户清楚页面/视图及它们的目的

重要的信息经常被隐藏，反而不重要的信息占据了页面。还有一些情况下，用户对应用程序的使用目的是什么完全毫无头绪。因此，应提供必要的信息并合理编排信息，以便使用户首先能够明确相关的东西。这样，用户才能通过屏幕表现知道改做什么和如何做。

使用准确的表述和说明——不要过于简化（特别是在网络应用程序中）

控件的可访问性

总体说明

这一页向那些应用 Web 开发控件,并希望他们的 WEB 应用易于实现的应用开发人员提供了的总体说明。详细的控件信息请参阅可实现性部分的内容。

说明

Web 开发控件的中心解释引擎已经提供了对于控件的总体描述，例如：类型，状态和屏幕显示文本。因此，应用开发人员只需要补充一些用户需要的特殊说明和描述。应用开发人员写的描述内容可以被添加到中心描述机制的缺省描述中。

例如：

如果一个 button 能够打开一个新的窗口，那么它的功能描述就必须被扩展。

通常情况下，如果一个 button 引进一个交互操作且这个过程不能被普通用户（blind user）认知，那么这个功能的描述就需要被扩展。

键盘可达性

因为应用开发人员不能直接设置 HTML 属性，他们无法设置元素的 tabIndex 属性。。

输入（input）元素和相应的标签

输入元素，例如：checkbox, dropdown listbox, input field, radio button, 和 text edit 控件等需要被关联到一个标签，因此用户就能知道标签与这个输入元素的联系。应用标签控件就是基于这个目的。

当应用键盘或鼠标时，一个标签和它所关联的输入元素同样说明了元素间的交互作用。

9. 布局

9.1 总体布局策略

这部分描述了 WEB 页，应用和视图的总体布局策略。一个页面的布局不仅仅是在页面内“扔”控件。还需要考虑如下的因素，如：

控制的流程——当用户工作时他会怎样在页面上进行操作。

依赖关系——页面上的元素是如何相互影响的

归属关系——在页面上哪些元素可以归类为同一组，在这些元素中，元素间的关系有近有疏。

审美与完整性原则——如何能真实有效地交流信息

在布局过程中有如下三步——它们可以按如下顺序进行： 要考虑到

- 元素的顺序（垂直的和水平的）
- 元素的嵌套
- 在不同层级的元素间的间距

顺序体现了流程的控制，依赖性和元素归类的信息——后者更具有线性的流程。嵌套也体现了依赖关系和从属关系——但体现在了层次中或上下结构中。间距体现了美学和合理的应用的完整性原理（多数体现了从属关系的特性）。

指南中的布局部门结构

这部分涵盖了整体布局的各个方面，例如序列的作用，嵌套和间距。布局的层次结构涵盖了嵌套的详细信息，也就是说，包含了哪些对象必须在同一层上，哪些必须被嵌套的信息。页面的 Flow Layout, Grid Layout 和页面的间距（单一或分组的控件）内容涵盖了定义间距的细节。

总体页面布局设计的各个方面

顺序的角色

元素的顺序典型的应该由控制流程决定，也就是说，被用户执行其任务的流程所决定。一般的，一个任务流可能不会是线性的，或者是由于某些错误，用户不得不重新执行某一步。这里，页面的设计者必须找到一个“中立”的序列来适应大多数用户或实际情况的需要。

另外，按照惯例，像信息导航，会在元素的排列中起到很重要的作用。按照一般习惯，排列元素的典型序列是从左到右，自上而下，

就像信息导航的顺序。依赖关系也典型的参照这种方式进行交互的。“该做的要先做”的箴言，表达了我们应该遵守“自然规律”的原则。例如：当我们输入一个客户的地址，我们由他的姓名开始，这是余下信息中最主要的信息——我们并不从街道名或住所的号码开始，尽管由这些信息也可能推断出客户的名字。

绝大多数的设计者也许都会很自然的遵守这样的规则，尽管他们没有去考虑过。然而，当问题出现时，例如这个规则被打破，所有的依赖关系或流程都变成相反的方向。这种逆向问题的产生往往会给用户产生很大的阻碍。

页面设计的第一步是排列页面上的元素。我们可以先建立一个样例而不去考虑页面设计的具体细节，然后同用户一起测试沟通（例如仅仅是画在纸张上的样例）。

嵌套的角色

有两个基本的方法直观的说明了元素间的关系——嵌套和邻近（closeness）。邻近的意思是一些相邻的对象能够被意识到他们的关系比那些联系不密切地对象更为紧密。元素间的紧密性典型的组合用来说明依赖关系和流程控制。例如：你要在搜索区域先输入一个值（左侧）然后才能点击和它邻近的查找按钮（右侧）。

嵌套是用来描述更复杂的层次间的关系和对象间的依赖关系。嵌套也是一种对用户隐藏细节的方法，因为用户可以先处理较高层面上的对象，然后决定他们想要更进一步的关注哪一个对象。

使用嵌套能使页面比单一的元素排列更复杂，因为嵌套可能需要界定边框或者用其它方式来隔离视图，这也许会使页面的视野变得混乱。因此，嵌套规则必须应用在以避免过度复杂的页面为目标的基础上（参阅层次排版和分层控制）

间距的应用能够帮助减少页面的混乱性，但是往往需要提供更多的空间

嵌套同样可以被预先开发一个样例（纸张上、HTML 的样例）；在这里，样例可以包含较多的细节

间距的角色

间距对于有同属关系的元素间的互动起了很重要的作用，它同样影响了页面内容的可读性和用户对内容的识别能力。

一般来讲，应用开发人员不应该需要考虑间距的细节，也就是说，他们不用考虑类似像在一组的 **button** 与边框之间需要插入多少个像素这样的问题。有两个控件控件：**grid** 布局控件和流程布局控件。它们将负责提供合理的间距布局。另外，整个容器，像 **tray** 和 **group**，也会相应的控制外部的间距。

注意：目前，间距控件并不能像期望的那样工作，因此，开发人员应当参考 **grid** 布局和流程布局的限制范围。

只有高层次的并想要提供真实最终版本网页预演的样例才需要考虑到间距的细节。

只有高级别的原型，它倾向于提供一个真实的最终页面的预览效果，才需要使用空格进行边界设置

相关控件

Flow Layout, Grid Layout

9.2 布局层次

描述了网页的布局层次，它定义了嵌套页面元素的选项，简单地说，它将指导设计人员页面元素如何放置到页面容器中——包括将一个容器的元素放置在另一个容器的元素中。

对页面和页面中各个部分进行层次布局是建立文本布局的基本规则。其主要目的就是防止由于应用大量的嵌套而产生过于复杂并引起视觉混乱的页面。

注意：这些规则并不包括对在元素内和元素间的间距定义。

从容器到布局层次

页面元素既可以是容器型的也可以是非容器型的。容器能够装载其它的元素，非容器型的则不能。层次布局对基本的容器型元素的操作进行了如下的描述，也就是对那些可以包含其它元素或容器的元素进行描述。这个非常重要，因为过分的嵌套会导致页面表现的过于繁重。

应用型容器

在层次布局的最底层有个最“根”层次的容器，它包含了整个应用。在 Web 或门户型环境中，需要考虑两种情况：

应用容器是简单的后台程序。例如一个框架或是窗口。这是被称为 Web 应用包括门户(portal)管理应用的情况。

应用容器是一个盘子(tray)或瓷砖(tile)。容器可能会有自己的元素或者控件;装载在这种容器中的应用可能会使用容器提供的服务。从应用的角度讲,这种容器就是它全部的认知——至少是从设计的观点来讲。

应用程序中的容器控件

应用中,容器控件定义应用的布局层次.这种类型的容器包括:

区域(Areas)(仅仅指 Web 网络应用)

标签栏(Tabstrips)

组(Groups)

子组 (Subgroups) (有或没有标题的简单元素的组——不包括在组控件中)

概要解释——线性序列和容器序列

从技术的角度来看，并不是所有的这些容器都是“真正的”容器。Areas 是一个应用的一部分。也就是说，在一个应用中，areas 构成了一个线性的序列。子组(subgroups)是可能含有标题的简单页面元素的集合。特别的是，它们被空字符(white space)或分隔线从余下的页面中隔离出来。

从布局的角度讲，当然，将区域和子组(subgroups)看成是“真正的”容器会容易一些，它并没有影响到布局的整个过程。将这些元素看成是真正的容器的好处是，布局的风格可以是层次或树型结构的，这样就很容易的获得整个页面或者是应用的结构。

非容器

当容器为页面建立了“骨架”，非容器的元素就是组成页面的“肉”。这些元素是域，按钮，选择元素，文本单元和表。这些元素的复杂性、嵌套规则要保证页面不会过于复杂。例如，table 视图的复杂程度和组及标签栏 (tabstrips) 近似。因此，table 与那些容器都会在布局层次的同一层。一个有关这方面相应的规则描述为：如果 table 视图是容器中唯一控件，那么它们不可以被放置在组或标签栏 (tabstrips) 中。

分隔符

分隔符，例如线或是空字符 (whitespace) 用来“分隔”元素或者容器。因此，他们很难被集成到一个页面布局的层次模式中去。他们能够被看作是“结束性的元素”或者容器的“边框”（他们较容易被集成到一个“线性”的布局模式上）。

请注意分隔符之间是有区别的。当你想要用空字符 (whitespace) 来分隔在同一布局层次上的容器或者元素时，你最好选择用线来分隔，因为空字符 (whitespace) 会产生不必要的框架。

创建布局层次

布局层次就是在页面上设置容器和简单元素。如下规则规定了页面上的元素如何能够通过垂直或水平的顺序连接或者通过嵌套的方式被组合，容器可能包含容器（节点），元素（叶子节点），或者两种都有。另外，非容器型的元素可能会和容器型的元素在同一层上出现。但是它们是“终节点”，并且不会在延续布局的层次。

例如：一个 table 视图可能与组或者标签栏 (tabstrips) 在同一层上。

布局的规则规定如下：

容器可能包含其它的容器——容器要包含它本身

嵌套的特殊条件，例如，单独的或与其它的元素一起

嵌套会有多少层

某些简单元素如何放置到某些容器中——这种情况的特定条件是什么

某些容器和某些简单的元素如何放置在同一层次上

WEB 应用的层次布局

依赖于使用的容器元素，不同的应用类型可以有不同的布局层次。考虑到门户的环境，不同的应用类型可以使用不同类型的容器，考虑到不同的应用目的，因此需要在布局方面有所不同。

Web 应用

应用的背景=窗体/框架的背景=应用的容器

- 区域可能包含：

标签栏（tabstrips）可能包含：

组（如果它不是唯一的元素）

子组(subgroups)

table 视图（如果它不是唯一的元素）

简单的元素

分隔符

- 组可能包含：

标签栏（tabstrips）（如果它不是唯一的元素）

组（如果它不是唯一的元素并且只针对不同类型的组）

子组(subgroups)

table 视图（如果它不是唯一的元素）

简单的元素

分隔符

- 子组(subgroups)可能包括：

简单的元素

分隔符

table 视图

简单的元素

分隔符

通常来说，在 Web 应用中不能有多于一层的嵌套。同时要注意标签栏(tabstrips)不一定要嵌套。

简单的元素为：输入框，checkbox，文本框，button.等

嵌套常规

如下的嵌套规则基于上面阐述的 table 整体视图和对设计原则的整理，例如，防止过量的帧及避免多余的标题（header）。

避免多余的标题

为布局分层而定义的嵌套规则要尽量避免多余的标题。因此，

在标签栏（tabstrips）或 areas（仅对于 Web 应用）内不要只放置分组框（group boxes），在组控件中不要放置只有标题的表。

避免过度的使用框架（复杂的视图）

过度的使用框架及边框使页面看起来很复杂并且浪费页面空间。因此，

在组控件中不要放置只有标题的表——用表的表头来代替

在组控件中不要只放置标签栏——将它们放置在区域中；应用标题文本，或者应用区域标题作为标签栏的名称

在组控件中不要放置组控件——应用带有文本标题和分隔符的组代替（不强制但应该在应用的过程中注意到——尽量应用不同类型的嵌套）

在标签栏中不要放置标签栏——嵌套标签栏是隐藏信息的理想方法

分隔线不要放置在容器之间或近似于容器的元素之间

注意：在这里你所看到的为间距和布局设置的信息目前还不能应用于表格（Grid）布局控件。

9.3 组控件间的间距

这一页讲述组控件间的间距划分的具体内容。包括如下的问题：

Tray 中的间距 — tray 的边框和它包含的内容之间的偏移量

恰当间距的优点 — tray 背后隐含的偏移量和间隔之间分析估计

组之间的间距 — 主从组之间的间距。也就是说嵌套的组之间的间距

有标题和边框的组控件之间的间距 — 这包含了更复杂的控件和成组控件

组中元素间的间距 — 组中的偏移量，例如组中边框和内容，组标题和内容的偏移量

组的排列 — tray 中组间偏移量和组队列

软组（soft group）的间距 — 对不能应用像容器那样的组控件的分组间距规则

间距规则的简要介绍：

Tray 的边框/标题和内容的偏移量：5 个像素

主从组之间的间距：10 个像素

有标题和边框的组控件间的间距：10 个像素

组内的偏移量，例如，组边框和内容的偏移量：5 个像素

组内的间距：标题与内容之间：10 个像素，内容和 button 之间：10 个像素。

软组（soft group）的间距：水平的 15 个像素，垂直的 30 个像素

下面为这些情况列出了正反两方面的例子。

在 Tray 中的间距

* 度量单位为像素 (px)

图1a: 在EP50中tray的偏移量.

tray 的偏移量只在 grid 布局控件中被列出, 并且是 5 个像素而不是假定的 10 个像素。在 EP6.0 中 tray 的整个内容的偏移量将被 tray 本身列出。在 EP60 新的设计规范中, 设计规范将会被重新定义。新的 tray 的偏移量在下图中列出

3px

7px

5px

7px

图 2: 环绕着tray的偏移量的例子

尽管你并不总是在右侧需要一个偏移量，你必须为 tray 设置一个右侧的偏移量。当前窗口(portal)布局的 tray 的大小决定了是否需要偏移量。

要避免如下错误。因为没有设置偏移量而使元素的视图超出了 tray.

图 3:环绕tray的错误偏移量的例子

恰当间距的优点

我们对组和 caesuras 都应用偏移量。在上面的例子，tray 内容周围区域的 5 个像素的偏移量保证了 tray 的内容被认定在 tray 中 规律的停顿 (Caesuras) 将区域分开。它们强调每一个 area 的各个属性，例如组。

主从组之间的间距

组控件中“主组”和“从组”间的间距，例如：无边框和无标题组的间距应该为 12 个像素。

用户名

密码

☐ 保存密码

电话

12px

图 4: 从组间的间距

账号

名称

方向 ☐ 向上 ☐ 向下

☐ 区分大小写

描述

用户名

密码

☐ 保存密码

电话

Caesuras 清楚的强调了 3 个area, 3 个area被分为 3 个组。

图 5a: 从组间间距的例子

9.4 单个控件间的间距

输入区域组 | check boxes 组 | radio buttons 组 | 连续垂直混合表单元素

这一页具体描述了单个控件间的间距。对于在组控件之间的间距，请参阅组间的间距。

如下包含了控件信息：

输入区域组 – 这是在表单排列应用中最常用的情况

check boxes 组和 radio buttons 组 – 这些元素经常在组中被用于提供选择和选项

连续垂直混合表单元素 – 这种情况包含了结合被安排在一个纵向组中的不同输入元素；对仅有几个栏的情况，参照多栏的 check boxes 和 radio buttons 的间距。

简要介绍间距规则如下：

单个元素的垂直间距：对于输入和下拉列表，5 个像素；check boxes 和 radio buttons，8 个像素。

多列纵向的水平间距：15 个像素

标签与输入框的水平间距，标签列的宽度：最宽的标签和偏移量的宽度在 8 到 22 像素之间

标签与选择元素之间的间距：check boxes 和 radio buttons 的选项是 8 个像素

如下是针对这些情况的正反例子

输入域分组

在输入域保留 5 个像素的偏移量

* 度量单位为像素 (px)

图 1a: fields间的偏移量

在应用中，左对齐的区域有参差不齐的右侧边框是很普遍的风格。

图 1b: 有参差不齐的边框的区域

图 3a: 标签与fields的偏移量

调整区域大小是允许的。当然，我们也很难说明为什么一个显示生日日期的区域会需要多于 8 个字符。

一方面人们永远不可能预知区域标签的长度，另一方面也不能预知在一个片断中会出现多少个连续的作用域，所以很难在标签和输入域之间给出一个标准的偏移量。作为一个大体的规则：在输入域中的每一行都要兼顾其它行，并在最宽的标签和它本身输入域的偏移量中选择适合的间距。如果可能的话，尽量避免一个字符的偏移量小于 8 个像素，三个字符的偏移量小于 22 个像素。在一个片断中最宽的标签和其相应输入域左侧的偏移量为 8 个像素。

通过限制最宽标签和其相应内容的间距最大值，我们保证了最小标签和它相应的输入域的偏移量不会太大,并且用户可以有针对性的调整标签和输入域的大小。

在这里你还是可以调整最大标签和它相应的区域，但这样使调整“邮件地址”到它的输入区域的大小变得困难。

图 3c: 标签和区域间偏移量过大的例子

尽管所有的偏移量看起来都是对的，“Reenter Your Password”和它的输入框间没有偏移量，使得整个的界面看起来很难看。



图 3d: 标签与区域间偏移量过小的例子
check boxes组

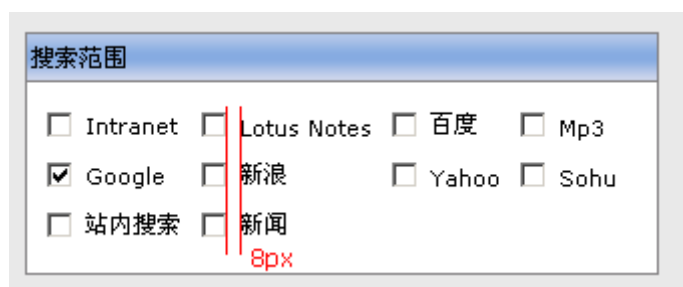
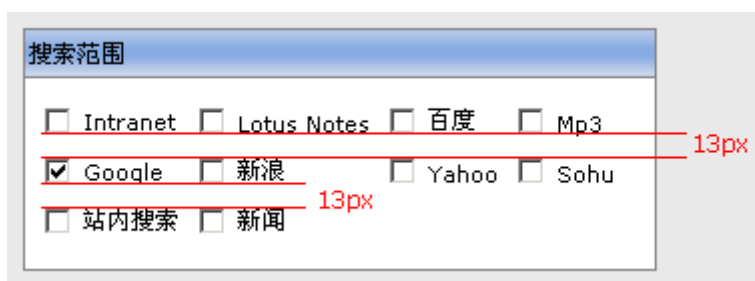


图 4a: check boxes和它们的标签间的偏移量

在每行 check boxes 间保留 8 个像素的偏移量



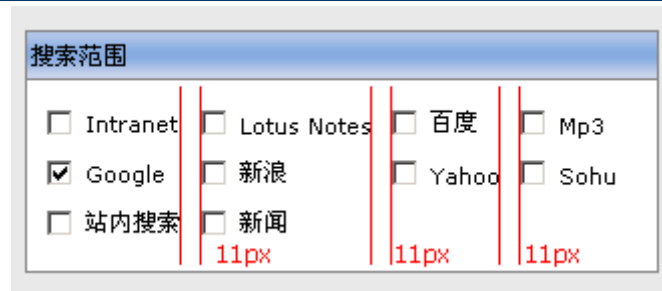


图 4c: check boxes组间的偏移量
Radio Button组

在 Radio button 和它们相应的标签间保留 8 个像素的偏移量

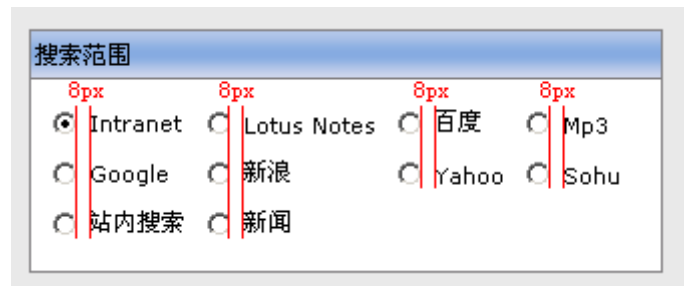
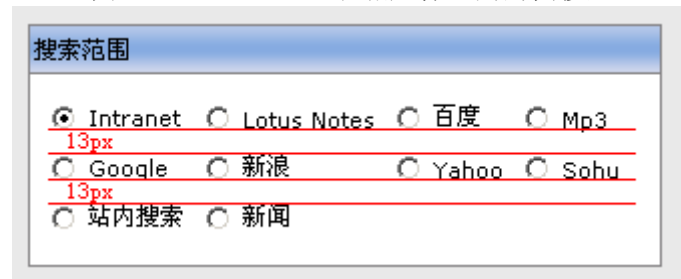


图 5a: radio buttons 和相应标签间的偏移量



在 radio button 的行与行之间保留 8 个像素的偏移量

图 5b: radio buttons行间的偏移量

在 radio buttons 的栏与栏之间保留 15 个像素的偏移量

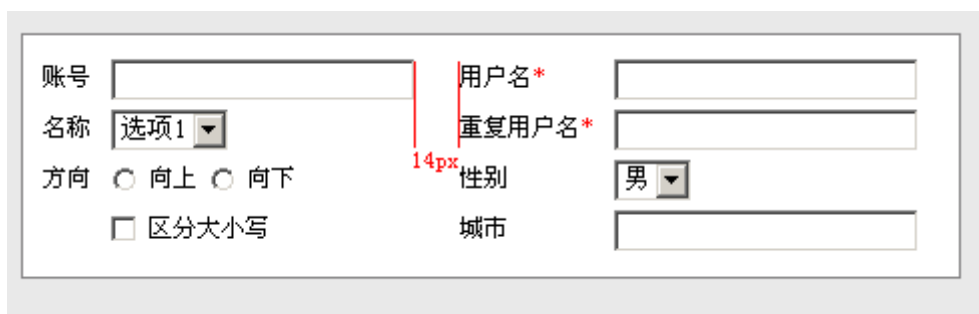


图 5c: radio button组的偏移量

纵向连续的混合型表单元素

在纵向连续的表单元素区域之间总要有 5 个像素的偏移量

The diagram illustrates a vertical form layout with the following elements and spacing annotations:

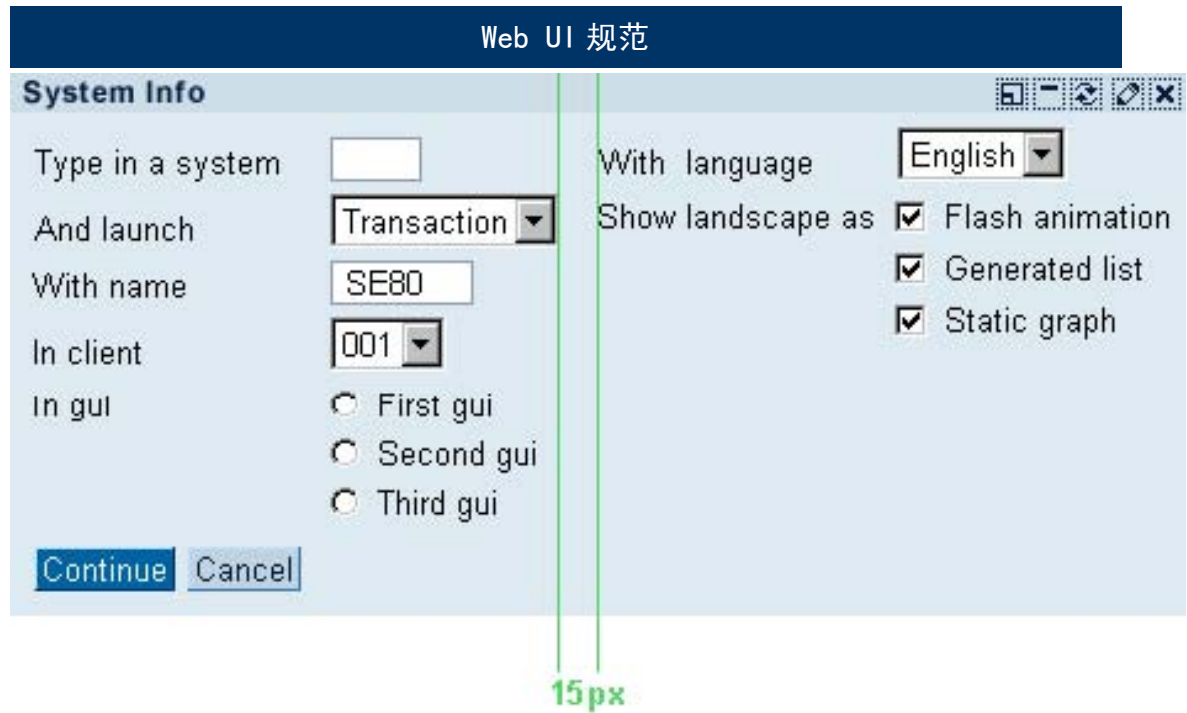
- Form Elements:** 用户名* (Username*), 重复用户名* (Repeat Username*), 城市 (City), 地址 (Address), 性别 (Gender) with radio buttons for 男 (Male) and 女 (Female).
- Spacing Annotations:**
 - Red lines and numbers indicate a 10px offset from the top and bottom edges of the form area.
 - A 4px offset is shown between the text labels and the input fields.
 - A 10px offset is shown between the form area and the right edge.
- Legend:** * 度量单位为像素 (px) (Unit of measurement is pixels).

图 6a: 纵向连续混合表单元素间的偏移量
在 button 上端或下端的表单元素间总会有 8 个像素的偏移量，不管它的前后是什么元素

The screenshot shows a 'System Info' dialog box with the following elements and spacing annotations:

- Form Elements:** Type in a system (text input), And launch (dropdown menu), With name (text input), In client (dropdown menu), In gui (radio buttons for First gui, Second gui, Third gui), With language (dropdown menu), Show landscape as (checkboxes for Flash animation, Generated list, Static graph), Continue button, Cancel button.
- Spacing Annotations:**
 - Green lines and numbers indicate an 8px offset between the 'In client' dropdown and the 'In gui' radio buttons.
 - Another 8px offset is shown between the 'With language' dropdown and the 'Show landscape as' checkboxes.

图 6b: 纵向连续混合型的表单元素间的偏移量



在表单元素的各栏间保留 15 个像素的偏移量

图 6c: 纵向连续混合型的表单元素间的水平间距



流程布局

用法 | 相关控件



图 1: 应用流程布局的例子



图 2: 流程布局中的控件



图 3: 控件会约束自身来适应它容器的大小（在这里是一个组）。

流程布局是一个用来一一结合其它控件的不可视控件。它能够被插入到任何容器控件中。

用法

如果你不需要在你的界面中对其它的元素进行对齐控制，你可以应用流程布局。如果在一行中提供显示所有控件的空间不够的话，在流程布局中的控件能够被隐藏。

要在流程布局（flow layout）中分离控件，目前，你应该使用包含一个简单的空字符的文本视图控件。

展望：为了简化和规范控件的分离我们将引进一个能够用来代替文本视图控件的控件分离器（separator control）

何时应用流程布局 - 何时应用 Form Layout

如果你不需要在你的界面中对其它的元素加以控制，你可以应用流程布局。这将提高性能，因为流程布局在排列时没有过大的 table 框架

在用户界面中应用表单布局（form layout）来控制其它控件的对齐。

相关控件

表单布局（form layout），grid 布局，文本视图

更多关于流程布局的信息

浏览器的兼容性 | 界面编辑器的可编辑性 | 可访问性 - 508 支持

浏览器的兼容性

flow layout 控件能够组织任何浏览器中的元素。

界面编辑器的可编辑性

目前，界面编辑器还不能改变流程布局

10. 布局控件

10.1 流程布局



图 1: 应用流程布局的例子

流程布局是一个用来一一结合其它控件的不可视控件。它能够被插入到任何容器控件中。

用法

如果你不需要在你的界面中对其它的元素进行对齐控制，你可以应用流程布局。如果在一行中提供显示所有控件的空间不够的话，在流程布局中的控件能够被隐藏。

要在流程布局（flow layout）中分离控件，目前，你应该使用包含一个简单的空字符的文本视图控件。

展望：为了简化和规范控件的分离我们将引进一个能够用来代替文本视图控件的控件分离器（separator control）

何时应用流程布局 - 何时应用 Form Layout

如果你不需要在你的界面中对其它的元素加以控制，你可以应用流程布局。这将提高性能，因为流程布局在排列时没有过大的 table 框架

在用户界面中应用表单布局（form layout）来控制其它控件的对齐。

相关控件

表单布局（form layout），grid 布局，文本视图

更多关于流程布局的信息

浏览器的兼容性 | 界面编辑器的可编辑性 | 可访问性 - 508 支持

浏览器的兼容性

flow layout 控件能够组织任何浏览器中的元素。

界面编辑器的可编辑性

目前，界面编辑器还不能改变流程布局

可访问性 - 508 支持

流程布局没有特定的增加可访问性的能力。它能包含一系列的自己在可访问的层次的控件，还可以提供更进一步的描述给初级用户

10.2 表单布局

图 1：三个表单布局都允许在显示方式的调整中编排以上表单元素和 button，详细的内容请参阅例子。

表单布局（form layout）是对应用容器、组或在某种意义上列表的其它容器在编排和对齐控制中的不可视控件。元素也可以在一个单元格中被预先封装。

表单布局（form layout）代替了之前的 grid 布局控件。

用法

应用表单布局（form layout）在容器中以列表（tabular）风格来进行对齐控制。尤其是，在组、标签栏和 tray(iViews)中应用表单布局（form layout）。由表单布局（form layout）来定义的 grid 是由带有单元格的行组成的。这样，grid 中的行和列被渐渐的定义出来。单元格可以跨距多个列。另外，在单元格中的元素能够被环绕。在单元格中可以添加多种控件。

你可以为在不同层次上安排页面元素而嵌套表单布局（form layout）。

何时应用表单布局（form layout）

- 在用户的界面中兼顾其它控件的同时，应用表单布局（form layout）来部署控件。应用表单布局（form layout）最普遍的是：(1) 列出容器中所有的表单；(2) 排列不同的容器或者表单布局。具体细节和两方面的应用见下例。

即使你不需要在界面中将控件和其它元素对齐，你同样可以应用表单布局（form layout）。在这种情况下，在每一行中只插入一个单元格并且设置这个单元格的宽度，这样，它就会超过表单布局的宽

度。

注意：表单布局（form layout）和 grid 布局类似，但 form layout 具有更多调节 grid 单元格的特性。你并不需要指定一个固定数量的行或列而只需要简单的在各行中添加行以及单元格。表单布局（form layout）依然可以在单元格中环绕元素但是在处理这种情况时流程布局会更加有效。

元素和间距的概述

表单布局包括了 3 种元素，每一种都有它自己的间距规范：

表单：底部空白，左侧空白，右侧空白，顶部空白（marginBottom, marginLeft, marginRight, marginTop）定义了表单布局的边框和它的内容区域之间以像素为单位的间距；每个页边空白的缺省值为 0。

行：底部填充，顶部填充（paddingBottom, paddingTop）定义了在行中的顶部与底部填充的像素值。缺省值为 0。

单元格：底部填充，顶部填充，左侧填充，右侧填充（paddingBottom, paddingLeft, paddingRight, paddingTop）定义了在单元格边框和它的内容间，以像素为单位的间距；缺省值为 0。

图 2 提供了一个表单布局中不同填充以及边框空白的整体视图。

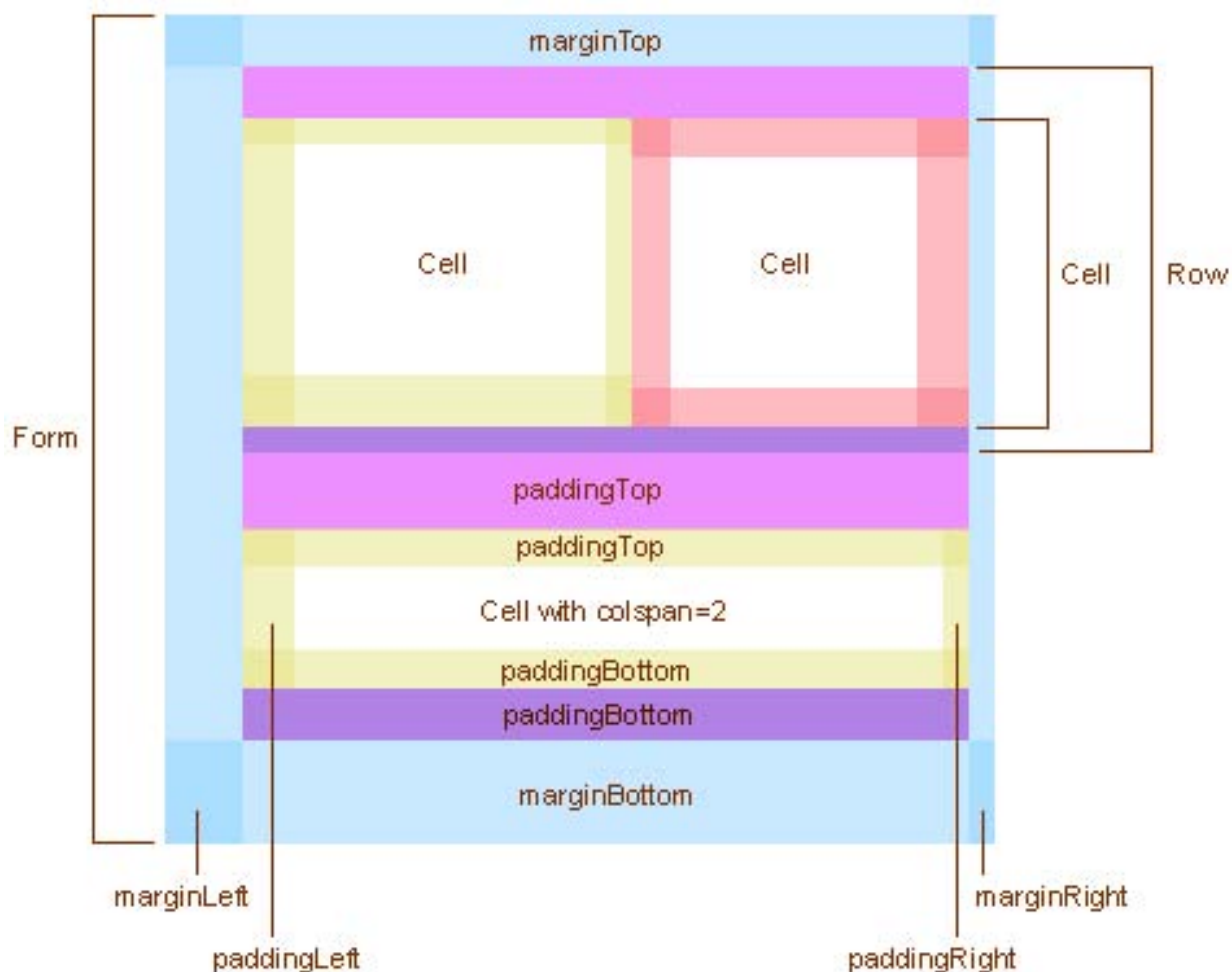


图 2: 在一个表单版面上不同页边空白和填充的整体视图

当在为表单布局添加元素时，注意要在不同层次上设置边缘空白区和填充的适当值。当嵌套表单布局时，为了避免额外的填充，在表单层上的边缘空白域必须在左侧，且缺省值为 0。细节请见下例。

例子

下列例子说明了如何应用表单布局。首先，确定在你的应用中需要被部署的不同区域。在图 3 的例子中有两个区域，一个表单区域包含标签和输入元素，还有一个按钮（button）行。两个区域分别应用表单布局来进行布局。在上部的表单布局中包括表单区域，也就是，标签和输入元素。第二个表单布局包括保存和取消按钮。两个表单布局都应用了另一个表单布局。

顶部表单布局

首先为输入元素创建一个 form 布局。在表单中设置所有的页边空白（底部空白，左侧空白，右侧空白，顶部空白）（marginBottom, marginLeft, marginRight, marginTop）为 5 个像素。注意组控件间的间距规范，设定了在 tray 的边框和它的内容之间的间距为 10 个像素。鉴于 tray 已经提供了 5 个像素的间距，表单布局（form layout）只

需要在适当的边框处提供另外的 5 个像素。

然后在顶部表单中添加 6 个行并且在除第五行外的每行添加两个单元格来实现两个列的排列。第五行只加一个单元格并且设置 `colspan=2`。

在单一控件的间距规范中规定，屏幕元素的每行间隔 5 个像素。鉴于顶部和底部的间距已经被正确的设置，除了最后一行的所有行设置底部填充为 5 个像素。

注意：在行与行之间可以采取其它方式设置 5 个像素的间距，但是，这样设置似乎是最简单的。

在设置标签和它们相应的输入元素之间的正确的水平间距时，不要指定左侧单元格的宽度。以设置左侧单元格的右侧填充 (`paddingRight`) 为 8 至 22 像素来代替。让其它的单元格填充保持它们的缺省值 0。

最后，在相应的单元格中添加标签和输入元素。

底部表单布局

为所有的 buttons 建立第二个表单布局并设置所有的空白区域为表单层的 5 个像素。因为在顶部表单布局中的底部有 5 个像素并且在第二个表单布局的顶部也有 5 个像素，你会在底部的按钮和输入元素之间自动的获得 10 个像素的间距。

在第二个表单布局中只为两个按钮添加一行两列。然后为每一个单元格添加按钮。

保留行和单元格层间的填充的缺省值为 0，除了左侧单元格的右侧填充。设置这个填充(`paddingRight`)为 5 个像素。这样保证了两个按钮之间恰当的水平间距。

The diagram shows a form layout with the following structure:

用户名	<input type="text"/>
密码	<input type="password"/>
语言	<input type="checkbox"/> 保存密码
描述	<div><div></div></div>
当前在线	<input type="checkbox"/>

Annotations on the right side of the diagram:

- Form Layout for the form with margins of 5px
- Row containing two cells - row borders not shown
- Row padding paddingBottom set to 5 pixels, except for the bottom row

图 3: 应用两个表单布局来部署控件
部署表单的布局 (Form Layouts)

因为两个表单的布局将要依次罗列出现，你需要在另一个表单内来编辑它们。这个外部表单的设计需要添加两行并在每一行添加一个单元格，它们将在一个列中（见（请参阅）图 4）。再把要部署的两个表单分别规划到单元格中。

注意这个外部表单布局（Form Layouts）一定不能有额外的空白区和填充。因此，让所有 4 个表单级的空白区保持它们的缺省值 0。同时，保持其它所有的填充值为缺省值 0。

图 4：用来编排两个垂直区域的另一个表单的布局

提示：你能够临时的设置 `debugMode` 属性的值为 `TRUE` 来实施框架的表单布局。这会使设置适当的布局变得容易一些。（请参阅表单布局（Form Layout）的详细信息）

设计相关属性

`form layout` 在表单级、行级和单元格级上具有与设计相关的属性，同时，在不同级的属性值之间还存在着依赖关系。请参阅图 2 以获得关于 `form layout`，以及 `form layout` 的元件和间距的概况。

表单级

`marginBottom`, `marginLeft`, `marginRight`, `marginTop`: 它们用来定义 `form layout` 的边缘和内容区之间的像素间距，每个值的缺省值为零。

`width`: 宽度的大小可以按像素来设定，或者按占容器的宽度的百分比来设定。如果单个单元格的宽度也被按照百分比设定，并且超过了 `form layout` 的宽度，单元格内容会被隐藏起来。

行级

• `paddingBottom`, `paddingTop`: 定义一行的顶部和底部填充像素，缺省值为零。

单元格级

`align` (`LEFT`, `RIGHT`, `CENTER`, `CHAR`, `JUSTIFY`): 在单元格内定义

元件的水平排列位置。

valign (BASELINE, BOTTOM, MIDDLE, TOP): D 在单元格内定义元件的纵向排列位置。

底部，左边，右边，顶部填充（paddingBottom, paddingLeft, paddingRight, paddingTop）：定义单元格边框和内容之间的像素间距，缺省值为零。

宽度（Width）：以像素，或者占 form layout 的百分比来定义宽度。注意：如果为一个列定义了不同的宽度，则使用最后定义的值，避免出现隐藏。同时注意，一个宽度值若超过 form layout，会导致隐藏行为。

colSpan：定义单元格在纵列上的水平扩展。

你也可以用布尔型 **debugMode** 属性来帮助完成一个恰当的布局。如果值为“true”，单元格会显示边框。

细节请参阅“Form Layout”

相关控件

Flow Layout, Grid Layout

关于 form layout 的更多信息

浏览器的兼容性 | 界面编辑器的可编辑性 | 可访问性- 508 Support

浏览器兼容性

flow layout 控件能够组织任何浏览器中的元件

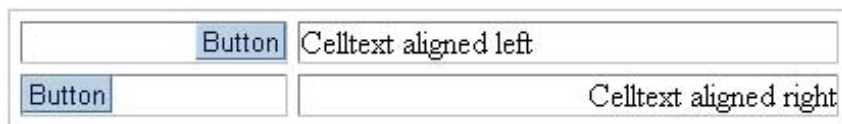
界面编辑器的可编辑性

目前，form layout 不能在界面编辑器内更改。

可访问性 - 508 Support

form layout 没有特殊的可访问性改进。它可以包含多个控件，而控件自身处于可访问的层级内，并提供针对初级用户更详细的描述。

结果



10.3 Grid布局

用法 | 样式相关属性 | 相关控件

用户名	<input type="text"/>
密码	<input type="password"/>
语言	<input type="checkbox"/> 保存密码
描述	<div></div>
当前在线	<input type="checkbox"/>

图 1: 控件的 Grid 布局安排

grid 布局是一种不可见的控件。在应用程序、组或者其它容器中，它可以帮助你以表格的形式排列和对齐控件。

用法

注意：不建议使用 grid 布局—可用 form layout 来替代

使用 grid 布局可以以表格的形式来把控件排列在容器内。grid 由排列成行和列的单元格构成。各种控件可被添加进单元格里。你可以把 grid 布局插入到任何容器控件内。尤其用于组、标记栏 (tabStrip) 和 tray (iviews) 中。你也可以为了安排不同级别上的页面元件，而嵌套 grid 布局。（请参阅下边的例子）。

最普通的关于 grid 布局的用法是，容器内的表单布局，和不同容器的排列。存在两个管理行或列之间间距的属性。cellSpacing 和 cellPadding。对于以上例子，推荐使用五个像素的 cellSpacing。对于缺省的布局没有必要使用 cellPadding。细节请参阅下边的例子。

注意：目前，在布局时还不能完全达到推荐的 paddings 和 spacings 的标准。这里给出的建议是使用最理想的布局的近似值。直到出现能够解决布局问题的新控件。

file:///F:/resources/htmlb_guidance/gridlayout.html (1 of 3) [17.02.03 10:27:29]

issues of the grid and flow layout controls.

grid 布局和流程布局控件的问题。

何时使用 Grid 布局-何时使用流程布局

在兼顾其它控件的同时，在用户界面内使用 grid 布局部署控件。注意：使用 form layout 代替 grid 布局。

如果你不需要在你的界面内调整控件与其它元件的布局，可使用流程布局。这样做可以提高性能，因为流程布局在表象上没有超越 table 结构。

例子

下边的例子示范如何使用 grid 布局。首先，确定出你的应用程序中需要排列的不同 area，在例子中，图 2 里有两个 area，一个表单 area 包含了必须被排列的输入元件，还有一个用来放置 buttons 的 button 行。

两个 area 由不同的 grid 布局组成。上边的 grid 布局包含表单 area，标签和输入元件。下边的 grid 布局包括“保存”和“退出” button。作为控件之间的间距，一般应该为 10 像素，设置单元格间隔为 5。

用户名	<input type="text"/>
密码	<input type="password"/>
语言	<input type="checkbox"/> 保存密码
描述	<div></div>
当前在线	<input type="checkbox"/>

图 2：第一步—内部控件的 grid 布局安排

这两个 areas 应该纵向相临显示。因此，你需要另外一个由一行两行组成的 grid 布局。在这个 grid 布局里只需要添加表单 area 和 Button 行（参阅图 3）。新的 grid 布局不需要额外的间距，因为这两个附件的 area 已经有了正确的间距。

用户名	<input type="text"/>
密码	<input type="password"/>
语言	<input type="checkbox"/> 保存密码
描述	<div></div>
当前在线	<input type="checkbox"/>

图 3:第二步—areas 的 grid 布局安排

提示：你可以临时设置 `debugMode` 属性值为 “true”，来使 `grid` 布局具有一个框架。这可以帮助你实现一个适当的布局。（细节请参阅 `Grid` 布局）

样式相关属性

你可以设置列的数量（`columnSize`），单元格之间的间距（`cellPadding`），和单元格内部的间距（`cellSpacing`）。

你也可以用布尔型 `debugMode` 属性来帮助完成一个恰当的布局。如果值为 “true”，单元格边缘显示。

详细资料参阅 `Grid` 布局页

相关控件

流程布局，`form layout`

关于 `Grid` 布局的更多信息

浏览器兼容性 | 界面编辑器的可编辑性 | 可访问性

浏览器兼容性

`Grid layout` 控件能够组织任何浏览器中的元件

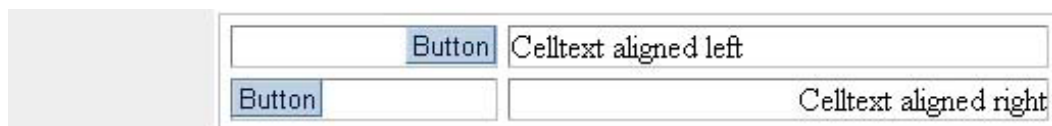
界面编辑器的可编辑性

目前，在界面编辑器内不能更改 `grid` 布局。

可访问性 - 508 Support

`grid` 布局没有特殊的可访问性改进。它可以包含多个控件，而控件自身处于可访问的层级内并提供针对初级用户更详细的描述。

结果



```
//>
```

```
<!--miaoql
```

11. 可视化控件

11.1 导航栏

用法 | 类型 | 样式相关属性 | 相关控件



图 1:使用导航栏（breadcrumb）搜寻结果

导航栏（breadcrumb）

告诉用户它们当前在层次中的位置，例如一个应用程序，一个目录，一个站点，或者是一个文档。

轻松返回起点，或者导航到层级中的其它层。

用法

应用程序软件和网络中的信息经常被组织成为层级的形式；总体信息会具体成细节信息，因此要创建一个信息层。导航栏（breadcrumb）控件会告知用户这个层级中有效信息的路径。例如：“查询目录”，典型的体现了一个导航栏（breadcrumb）把层级位置作为查询结果告知用户，把用户引向目标条目。如果导航栏（breadcrumb）在路径描述时使用链接，用户就可以跳转至特定的文件夹或主题。

处在导航栏（breadcrumb）链接中的某一项称为“breadcrumb item”，“breadcrumb item”可以通过模型或者手动定义。

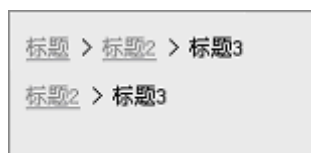


图 2:折行的导航栏（breadcrumb）

如果导航栏（breadcrumb）这一行变得长于浏览器窗口的宽度，导航栏（breadcrumb）就会像文本行一样换行显示。换行时在单词的结尾位置分开，例如空格位置。如果在导航栏（breadcrumb）字符串中没有分开的单词，会在导航栏（breadcrumb）的分隔符“>”后边换行。

类型

导航栏（breadcrumb）可以作为一个简单的路径信息（无链接，图 3 顶部），作为一个层级内部的可点击位置，或者作为一个被路径信息描述的链接（图 3 底部），显示出来。

FirstPathItem > SecondPathItem > ThirdPathItem

[FirstPathItem](#) > [SecondPathItem](#) > **ThirdPathItem**

[FirstPathItem](#) > [SecondPathItem](#) > [ThirdPathItem](#)

图 3: 简单路径信息（顶部），每个“breadcrumb item”是个独立的链接（中间），整个路径是可选择的（底部）。

导航栏（breadcrumb）类型通过属性“behavior”来设置：值

“SINGLELINK” 会创建一个全部路径都是可选择的导航栏 (breadcrumb)。

值 “DEFAULT” 创建一个每个项都可被独立的链接的导航栏 (breadcrumb)。

用法 - 类型

为以下目标使用不同类型的导航栏 (breadcrumb)

使用路径信息导航栏 (breadcrumb) (图 3 顶部) 来指定层级中文件的位置。例如：一个查询结果的目录不仅显示查询结果本身，而且还有这些结果的路径。

如果你允许用户们在层级内上下移动，或者跳转至某个范畴，可以使用独立链接导航栏 (breadcrumb) (图 3 中间)。导航栏 (breadcrumb) 的最后一项往往显示的是页面，而它不是一个链接。

使用全部导航栏 (breadcrumb) 路径作为单独链接 (图 3 底部)，来告知用户链接目标在层级中的位置 (实际上就是导航栏 (breadcrumb) 的最后一项)。

样式相关属性

Size 导航栏 (breadcrumb) 的字体大小有三种：large, medium (缺省值) 和 small (图 4)。为属性 “size” 设置值 LARGE, MEDIUM, 或者 SMALL。

FirstPathItem > SecondPathItem > ThirdPathItem

FirstPathItem > SecondPathItem > ThirdPathItem

图 4: Large, medium, and small size breadcrumbs

使用符合当前文本环境的文本字体大小。在空间比较紧凑的环境下尽量使用小字体。这些字体也可用于式样和突出显示的场合。

相关控件

Links

浏览器兼容性

在 Netscape 4.x 和 6.x 浏览器中，导航栏 (breadcrumb) 路径图标，并不是粗体显示的，只是普通的字体粗细。

Text 1 > Text 2 > Text 3

图 1: Netscape 浏览器路径图标具有普通字体粗细

界面编辑器的可编辑性

在界面编辑器中， 下列 tree view 控件中的属性可以被更改。

IE 5 系列和 Netscape 4.7 以上版本

Breadcrumb Styles 导航栏（breadcrumb）样 式	Font Color of Breadcrumb Path Icon	x	x
	导航栏（breadcrumb）路径图标字体 颜色		
	Font Weight of Breadcrumb Path Icon	x	x
	导航栏（breadcrumb）路径图标字体 粗细		
	Text Decoration of Active Entry	x	x
	活动窗口的文本修饰		
	Breadcrumb Padding	x	
	导航栏（breadcrumb）填充		

Table 1:对于树型视图控件的可编辑格式。获得普通样式参阅
“Customer Branding and Style Editor”一文中的 HTMLB 控件和界面编辑器部分。

可访问性

操作（Keyboard）：若导航栏（breadcrumb）控件包含链接，它将以缺省的方式插入到可访问的层级。

缺省描述：由 HTMLB 解释引擎来提供

应用细节描述：如果需要的话，使用“setTooltip”方法设置一个附加的描述。

用法 | 类型 | 样式相关属性 | 相关控件



11.2 按钮

用于实现一些外部功能例如指向给定目标，或者实现导航目的

图 1:一个本身含有两个 Button 的 ivew，并且还包含带有 button 的组。

用法

button 仅用于少数而且很重要的功能上。应用大量的 **button** 会使屏幕看起来沉重而且复杂。因此，当对是采用链接还是 **button** 产生怀疑的时候，还是用链接吧。因为链接的视觉效果和视觉优势，使之更适合设计目的。

注意：关于何时使用 **button** 和何时使用链接的细节，参阅“链接”。

按下回车键激活缺省功能（主要的，但不是必需的，与强调 **button**（**emphasized button**）的功能一样）对于不同类型和大小的 **button** 的使用方法指导将在下面给出。

标注

使用标题格式定义 **button** 的标签，在 **button** 标签上使用省略号("...")来指出这个命令需要补充信息才能执行。特别是当为用户提供一个对话框来输入缺少的信息时要使用这种方式。

注意：标题方式意味着，每个单词的首字母要大写，除了确定小写的词，例如条款和短语前缀。

谨慎的选择 **button** 的功能描述；试着使之尽可能的清楚。对于复杂的交互，用动词与名词的结合描述。例如，“数据库检索”。如果上下文比较清晰，也就是，如果动作仅仅应用于一个对象，那么用一个单一的动词（“检索”）作为 **button** 的标签就足够了。对于 **shufflers** 和要进行比较的元件实现的功能，你可以使用一个简单的“Go”作为 **button** 的标签。

位置与图样选择

button 位置布局规则详细请参阅“**button** 位置布局”。

button 与链接在功能上相似。何时用 **button**，何时用链接的讨论，参阅“链接”。

用法—Disabled button vs 隐藏 button

Disabled button 指示一种功能不可用。因此，为暂时不可用的功能使用 **Disabled button**。举例，在某个系统状态下，例如一个“错误”**button**，可以防止用户执行某种功能。

Hide button 对用户来说永远是无效的，例如，用户不会获得执行某个动作的许可。

相关控件

链接，输入域，组，Table 视图

关于 **button** 的更多信息

浏览器兼容性 | 界面编辑器的可编辑性 | 可访问性

浏览器兼容性

在 Netscape 4.X 浏览器中 button 会作为标准 HTML button 显示



图 1: Netscape button

在界面浏览器中的可编辑性

在界面编辑器中，button 控件中如下的属性值可以被更改（Internet Explorer 5 及以上版本中，Netscape 4.7 中使用标准 HTML buttons）

Group Style IE 5 及以上版本

button 样式：文本填充 X 文本装饰 X 字体粗细 X 边缘宽度和风格 X 标注 button：标准背景颜色 X 标准边缘颜色 X 标准字体颜色 X 标准旋转颜色 X Disabled 标准背景颜色 X Disabled 标准字体颜色 X 强调 button：强调背景颜色 X 强调边缘颜色 X 强调字体颜色 X 强调旋转颜色 X Disabled 强调背景颜色 X Disabled 强调字体颜色 X 标准大小 button：高度 X 小号 button：小高度 X 文本换行：白空格 X

Table 1: button 控件的可编辑样式

获得普通样式参阅“Customer Branding and Style Editor”一文中的 HTMLB 控件和界面编辑器。

可访问性

操作(Keyboard):button 以缺省方式插入可访问层级中—包括 button 状态（例如 Disabled 状态）和类型（例如强调 button）

缺省描述：由 HTMLB 解释引擎提供

应用细节描述：如果需要，使用“setTooltip”方法设置附加的描述。若用户需要更多的信息或者使用说明，就需要附加描述。通常情况下，一个 button 的介绍和交互使用若得不到初级用户的认可，那么描述就要被进一步扩充。例如：如果点击 button 会打开一个新窗口，描述信息就需要被扩展。

11.3 图表

用法	类型	样式相关属性	相关控件
		3.0	
		6.0	
		5.0	

用法

图表以图形的方式显示与用户相关的数据，以至于用户可以很容易的捕获数据本身的特性及数据间的关系。

有时对用户来说，掌握隐藏在数据背后的价值十分重要，从另一个角度会在 **table** 中以数字或者文本的方式显示数据。应该设置一个 **button**，允许用户在图解和 **table** 视图之间切换。

注意：关于图表和它的使用方法的完整信息参阅 SAP Design Guild 的 Recommendations for Charts and Graphics。

布置

图表可以在一个 **iView** 中显示，这样的话，它应该组成 **iView** 的主要部分。一个图表可以同其他屏幕元件联结起来，允许它们进行交互。

Legend

legend 说明了在图表中使用到的颜色。对于图表控件来说，**legend** 能够自动被产生，它可以被放置在图形的右边或者下边；其它位置也可能，但是应该避免。**legend** 的位置由属性 “**legendPosition**” 来设定，“**legendPosition**” 的值可以是 “**EAST**”，“**NORTH**”，“**SOUTH**”，“**WEST**”，或者 “**NONE**” 代表没有 **legend**。

屏幕元件的次序

如果还存在更多的交互元件，遵循以下规则：

- 如果数据可以从若干数据集中被提取，或者数据的数量必须被减少的话，可以在图表的上边放置一个 **shuffler** 过滤器。
- 然后紧接图表，参考图表的格式，在图表的右边或者下边放置 **legend** 或者其它文本。（如上）
- 为图表关联功能和图表下边的状态信息（例如图像缩放）放置 **pushbutton**，并把这些 **button** 左对齐。
- 若另外存在一个 **table** 视图，则放一个 **button** 在 **table** 视图下，可以使图表和 **table** 视图之间相互转换。

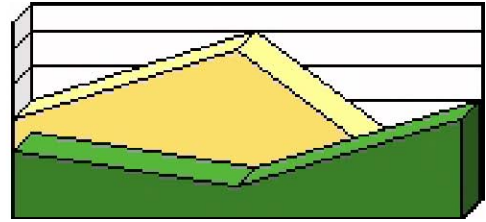
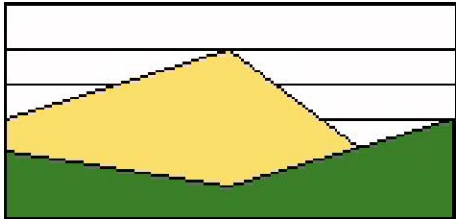
功能性

图表可以提供的典型的功能是：

- 在图和 table 之间切换
- 缩放和旋转（对于图表控件不可用）
- 向下钻取（Drill-down）

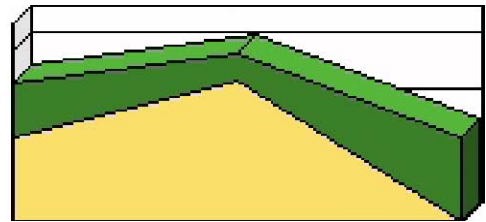
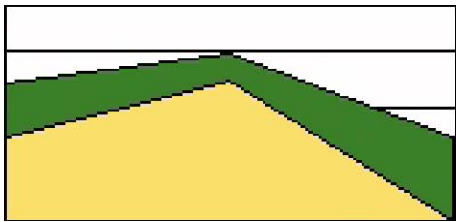
类型

图表可以有很多类型。由“chartType”属性来确定。以下是可用的图表类型及相应 chartType 属性值的总结。



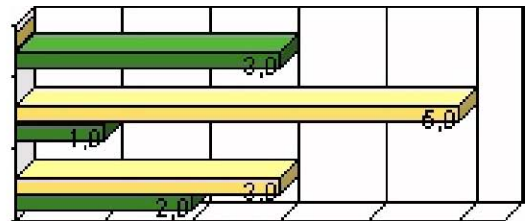
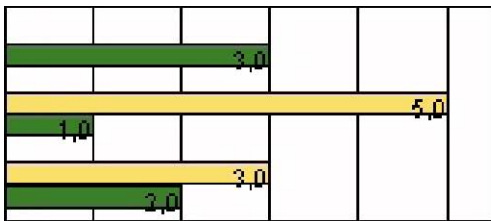
Area 图表(chartType = AREA)

三维立体 Area 图表 (chartType = AREA_3D)



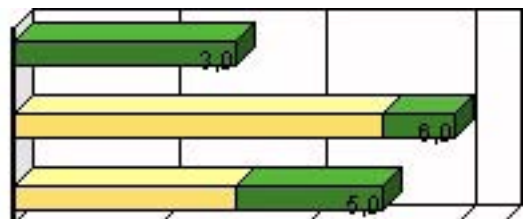
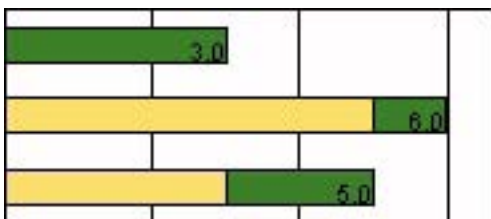
堆栈 area 图表 (chartType = AREA_STACKED)

立体堆栈 area 图表 (chartType = AREA_STACKED_3D)



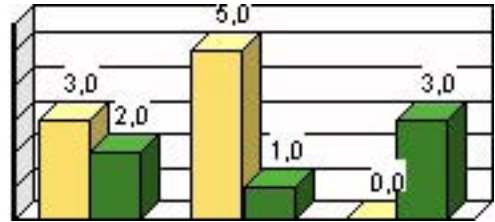
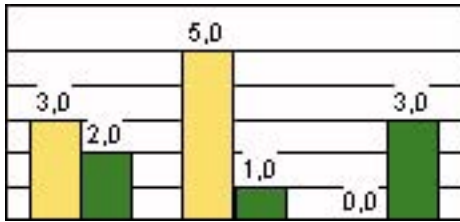
条形图表 (chartType = BARS)

立体条形图表(chartType = BARS_3D)



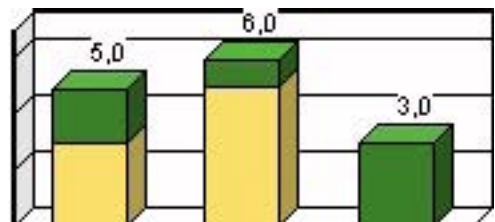
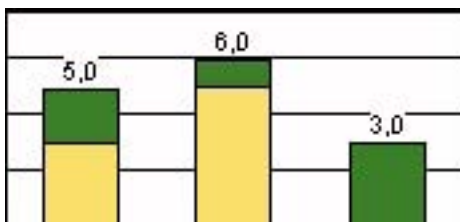
Stacked bar chart (chartType = BARS_STACKED)

Stacked 3D bar chart (chartType = BARS_STACKED_3D)

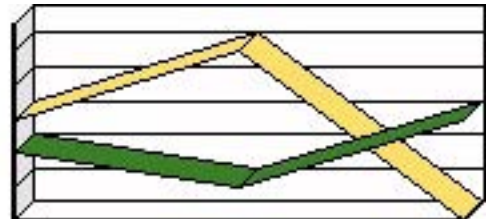
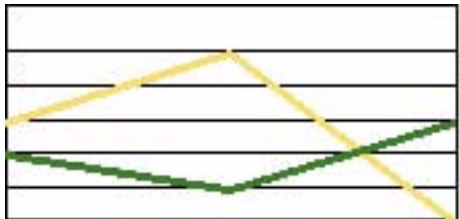


Column chart (chartType = COLUMNS) 3D

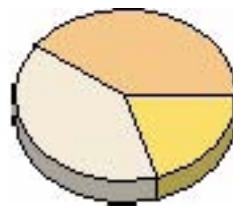
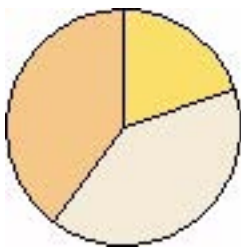
Column chart (chartType = COLUMNS_3D)



Stacked column chart (chartType = Stacked 3D column chart
(chartType = COLUMNS_STACKED) COLUMNS_STACKED_3D)



Line chart (chartType = LINES) 3D line chart (chartType = LINES_3D)

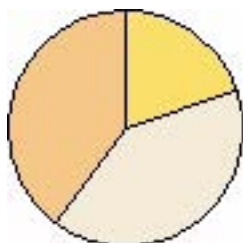


3D pie chart (chartType = PIE_3D)

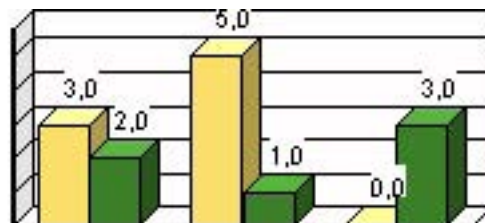
Pie chart (chartType = PIE) Extruded pie chart (chartType = PIE_EX)



Extruded 3D pie chart (chartType = PIE_EX_3D)



Bitmap chart (chartType = BITMAP) - arbitrary bitmap



Split pie chart (chartType = PIE_SPLIT)



Pyramid chart (chartType = PYRAMID) Trend chart (chartType = TREND)

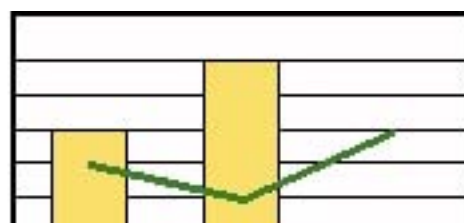


图 2:图表类型的概况

使用图表类型—概述

下列的 table 总结了当前可用图表类型的用法提示。

图表类型 典型应用程序的变量，注释。

Area 随时间变化的累积量（数值或者百分比，累计百分比）。

Column/Bar 随时间变化或者沿着纵向（Column）横向（bar）观察，不同情况下，数据组一定会表示为大量的 columns/barscolumns/bar 都是从零点开始的。

Segmented Column/Bar 随着时间延续，比例关系也许能达到百分之百

线，曲线

趋势，函数关系

数据点由线联结而成
order curves
有序曲线

饼状图 为了重点显示在一个片断点上的比例关系，这部分可以从当前饼状图中被突显出。（exploded pie chart）

Table 1: 图表类型以及它们的应用和变量

式样相关属性

图表控件的外观和行为可以由若干属性来控制。

legend 的位置和可见性：属性“legendPosition”允许隐藏，显示和设置 legend 的位置（值包括 NONE, EAST, NORTH, SOUTH, WEST）。

颜色次序：使用属性“colorOrder”来控制颜色的次序（值包括 DEFAULT, STRAIGHT, REVERSE, 和 SNAKE）。

高度和宽度：属性“height”和“width”允许设置图表的大小。

值与标题的显示：在你的配置中存在一些属性，通过属性值和目录（category）来控制外观和位置。更多信息请参阅图表页。

关于属性的细节信息，请参阅图表页

相关控件

图像，Table 视图

关于图表的更多信息

浏览器兼容性 | 界面编辑器的可编辑性 | 可访问性

浏览器兼容性

目前没有发现任何问题—图表类似于图像

界面编辑器的可编辑性

用户不能通过界面编辑器定制图表。界面编辑器没有提供能让图表象目录一样放置的可编辑样式。

可访问性

图表类似于图像，因此，相关的标准适用。

操作（Keyboard）：图表缺省时没有被插入到可访问层级。

缺省描述：由 HTMLB 解释引擎提供。

应用细节描述：如果需要的话，使用“setTooltip”方法设置一个附加的描述。不要使用“setAlt”方法设置备用文本（alt 属性）。

11.4 复选框

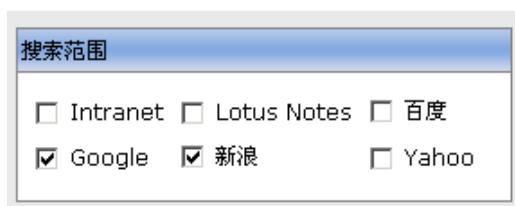


图 1: 一个checkbox组

checkbox 为用户提供了一个或者多个选择。用户可以选择零个，一个或者在 checkbox 组中所需要的多个选项。

用法

当用户可在多种选项中选择的时候，checkbox 是个合适的元件。它可以被单一显示或者成组的显示。

在 checkbox 组中，选择不是唯一的，用户可以在一组中选择多个选项。如过你需要应用单选，你可以用“radio buttons”或者下拉列表框来代替。

复选框组

对于使用很多 checkbox，如果可行的话，使用 checkbox group 控件。这个控件允许把 checkbox 按照列，行，或者以矩阵的方式排列。

注意：在 checkbox 组中是不可能更改水平间距的。如果你需要使用与 checkbox 组控件所提供的不同的间距，应该使用单一 checkbox 和可用的 grid 布局控件。

排列和式样选择

需要排列和式样选择的细节，参阅“Forms - Using Checkboxes”。

样式相关属性

checkbox 具有“disabled”和“checked”两个属性，如果一个 checkbox 不能被检验，或者不能被一个临时用户检验，设置“disabled”属性为“TRUE”。设置“checked”属性为“TRUE”，把 checkbox 设置为检验状态。使用“text”属性来为 checkbox 设置描述标签。

你也可以为 checkbox 组设置列的数量（属性 columnCount）。

相关控件

Radio Button，下拉列表框，列表框，标签，Grid 布局

关于复选框的更多信息

浏览器兼容性 | 界面编辑器中的可编辑性 | 可访问性

浏览器兼容性

checkbox 可以在每个支持它的浏览器中显示

界面编辑器中的可编辑性

checkbox 自动作为标准浏览器控件运行。界面编辑器的更改可以在相应的标签显示。

复选框组

checkbox 组在界面编辑器中不可编辑。

可访问性

checkbox 必须同标签控件联合使用，是指那些在 checkbox 左侧使用了标签的 checkbox。这保证了，访问页面的人能够明白元件之间的关系，并且能够理解 checkbox 的正确意思。

操作：（Keyboard）：checkbox 缺省插入到可访问的层级。

缺省描述：由 HTMLB 解释引擎提供

应用细节描述：如果需要可以使用“setTooltip”方法设置一个附加的描述

标签：必须与同一个用于左侧标签的标签控件相联接。（使用“setLabelFor”方法，来识别相应的 checkbox 或者 checkbox 组）。

挑选适当的选择控件

下拉列表框在功能上与 List Box 很相似，都为用户提供了一个可从中进行选择的条目目录，就是说，它们都是单选目录。

注意：如果要实现很小的条目数量间（2-6 个）的选择且用户需要立刻看到所有的可选项，应单独使用“radio button”。

11.5 日期导航

日期导航
用法 | 样式相关属性 | 相关控件



图 1:日期导航显示一个月的例子

用法

日期导航是对于各种动作进行高级处理的控件，这种处理需要输入并显示日期。因此，日期导航控件的主要目的是帮助用户键入日期，它也确保了日期被按照适当的格式输入。在如下的例子中，日期导航

强烈建议让用户选择日期的方式和允许用户输入日期的方式并存

注意：如果日期必须以特别的形式输入，那么在输入域的后边应当给出一个样例。

日期导航也能用来显示阳历的日历。

样式相关属性

日期导航允许设置每行的月份数（monthsPerRow）和每列的月份数（monthsPerColumn）。

相关控件

目前没有相关控件

关于日期导航的更多信息

浏览器兼容性 | 界面编辑器中的可编辑性 | 可访问性

浏览器兼容性

Netscape Navigator 4 浏览器不能显示标准日期导航控件的某些可视外观。



图 2:标准日期导航的例子

界面编辑器的可编辑性

在界面编辑器的 5.0 版本中，日期导航被称为“日历”。在界面编辑器中，可以改变背景颜色，文本属性，填充和显示在可点击元件之外的指针类型。以下是可以更改的界面属性列表。

Group	Style	IE 5 and above	Netscape 4.7
Day Names	一周中每天的背景颜色	x	x
Day Numbers	输入框的宽度	x	
	录入信息的对齐位置	x	
	录入文本的样式	x	
	背景颜色1.2的选择	x x	x x

Table 1:日期导航控件的可编辑界面。

可访问性

目前日期导航器还不能支持屏幕读取，获得更多的关于可访问性的信息，参阅“SAP Portals Accessibility Guidelines”。

11.6 下拉列表框

下拉列表框

用法 | 样式相关属性 | 相关控件

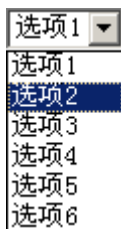


图 1: 一个具有六种语言条目的下拉列表框。

下拉列表框是在右侧带有箭头图标的输入域。点击这个图标会立刻在输入域下方下拉出一个目录，并告诉用户哪些值可以被选择。其中条目称为-List Box 条目。下拉列表框是只读的。

用法

使用下拉列表框

支持有限的选择值。条目的数量不应该超过 20。

能够在大量数据视图间跳转，尤其在“iViews”中，这是个节省页面空间的好办法。有了下拉列表框，更多的视图之间可以互相比对，例如在标记栏（tabStrip）中，因为视图中列出的项的数量不受空间的限制。但是，下拉列表应该不长于 12 个条目。

为了获得精简的数据视图，存在一个过滤大量数据集的过程。这个过程模仿自然语言的语句来叙述查询，也能仅使用查询语句组成语言。查询语句是典型的由动态元件（例如下拉目录，编辑域和选择元件）结合的静态文本组成的。

注意：下拉列表框控件不能自动添加描述标签。使用标签控件来添加一个描述。

下边的 table 为上述的用法描述举例。

Table 1: 下拉列表框的用法举例。

挑选适当的选择控件

下拉列表框在功能上与 List Box 很相似，都为用户提供了一个可从中进行选择的条目目录，就是说，它们都是单选目录。

注意：如果要实现很小的条目数量间（2-6 个）的选择且用户需要立刻看到所有的可选项，应单独使用“radio button”。

样式相关属性

下拉列表框可以被置为“enabled”或“disabled”两种状态。设置属性“disabled”值为 FALSE 来激活一个 checkbox，设置为 true 来使 checkbox 关闭。

激活下拉列表框的例子，

一个 Disabled 的下拉列表框不可点击，无条目可选。



图 2:Disabled 下拉列表框： Disabled 下拉列表框没有宽度属性。

注意，此控件采用最宽的条目的宽度值，该值被设为下拉列表框宽度值。

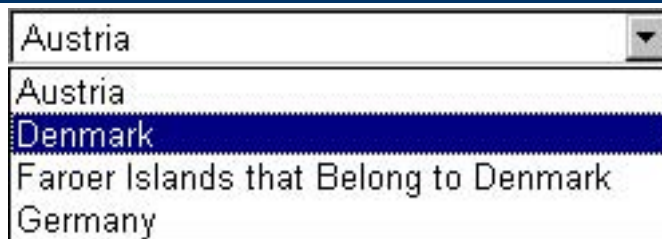


图 3:包含一个宽条目的下拉列表框

用法—Disabled 状态

若不允许用户改变下拉列表框的值，或者一组输入元件其中包含一个 Disabled 的下拉列表框，设置为 Disabled 状态。

例子：一组包含了一个由于用户未选中某个选项，而未被激活的下拉列表框的 fields。（参阅图 4）



图 4:Disabled 下拉列表框—因为用户选择了invoice选项，所以选择域未被激活。

相关控件

输入域，条目目录，标签，List Box，Radio Button,树状视图

关于下拉列表框的更多信息

[浏览器兼容性](#) | [界面编辑器的可编辑性](#) | [可访问性](#)

浏览器兼容性 Netscape 4.7

Netscape 4.7 中不支持 disabled 的 dropdown list box

浏览器 Netscape 6.1 和 6.2 版本

在浏览器 Netscape 6.1 和 6.2 版本中，下拉列表框看起来与标准控件有些不同。查看标准下拉列表框[点击这里](#)

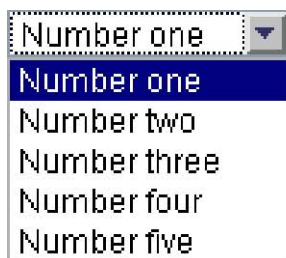


图 1: Netscape 6.1/6.2中激活的下拉list例子



图 2: Netscape 6.1/6.2中Disabled

界面编辑器的可编辑性

在界面编辑器中，下拉列表框不在自定义的元件列表中体现出来。此元件没有特殊的控件界面，仅提供通用界面。

可访问性

下拉列表框必须和指向下拉列表域的标签元素共同使用。这保证了，用户能够知道两种元素的关系并且可以依照下拉列表框的位置来找出相对应的标签

- 操作（Keyboard）：下拉列表框缺省插入到可访问的层级
- 缺省描述：由 HTMLB 解释引擎提供
- 应用细节描述：如果需要，使用“setTooltip”方法设置一个复加的描述。

标签：必须同一个标签控件联接使用（使用“setLabelFor”方法识别相应的下拉列表框）

11.7 文件上传

文件上传

用法 | 相关控件

文件上传控件是一种允许访问客户端文件并将其上传至服务器的控件



图1:在一个窗口中的文件上传的例子

用法

在你想要提供将文件传输至服务器的功能时，使用文件上传控件。

相关控件

导航栏（breadcrumb）, button

关于文件上传的更多信息

浏览器兼容性 | 界面编辑器的可编辑性 | 可访问性

浏览器兼容性

文件上传是一种安全敏感度较高控件，因为它允许应用程序访问客户端的硬盘。由于这个原因，我们应用了没有修改样式的原始的浏览控件，浏览 button 作为一个平台级的标准控件出现。

界面编辑器的可编辑性

不可编辑

可访问性

被提供

11.8 组

用法 | 类型 | 样式相关属性 | 相关控件

组控件集成了一系列的控件或信息：它说明了那些部分是应该属于一个整体的并且把它们从其它内容部分分离开来。图 1（至左边）

仅由背景颜色区分主从组的类型，可以用把部分内容的突出显示的方法实现。

在整页的应用程序中，主从组类型可以应用在创建一个可以放置其他控件的 area。

用法

在全页面应用程序中，使用组控件来：

将一系列连贯的信息组成一组并同其他组分离

定义一个可以放置文本或者控件的 area

使某部分应用程序或者信息突出显示出来

在 iView 中，通常应该没有突出显示或者分离不同组的必要，因为 iView 精确度小而且简单。但是，在某些情况下使用组控件是有意义的。

突出显示（highlight）或者分离 iView 的一部分，能更好的说明它自身的架构。例如，要表现某个 button 是同整个应用程序相关还是只同一部分相关，使用大字体文本来突出显示（highlight）某个文本信息部分

一般应用技巧

只在其他分离信息或者域组的方法不奏效的时候，才使用组。Group box 看起来像 iView 的 tray 容器，可能会在视觉上混乱界面。更适合的是，在组元件中使用空格或者垂直分割线，这依赖于完全形态法则（Gestalt laws）。

提示：有时你在 iView 中并不是真的需要组，仅仅是想要创建一个更好的可视框架。避免误用组控件，使用文本视图控件给用户一个更好的关于内容的概括：为每部分创建一个文本标签并且在两部分之间添加一条空行来分开它们。

尽量不要嵌套组，把大组中的小组用线或者空格分开。如果你真的需要嵌套组，考虑使用不同的组类型。（如下）

布置

只有在标记栏（tabStrip），table 视图，树状视图与其他元件共同出现时，才被允许包含在一个组控件之中（详细信息参阅 Layout Hierarchy）。它们像一个单独控件一样被安置，不需要在它们与周围元件之间，进一步添加间隔，因为它们已经有了清晰的边缘和基本形状。

类型

取决于组包含的条目，你可以从提供的界面中选择，目前，有 5 种组的样式可用，它们设置使用“design”属性。

主从组

主组（属性 design = PRIMARYCOLOR）和从组（属性 design = SECONDARYCOLOR）允许通过一个单色调的背景来显现组。它们都适用于文本。带有白色背景颜色的控件，例如输入域和 Checkbox，在两种组中都显得很突出。

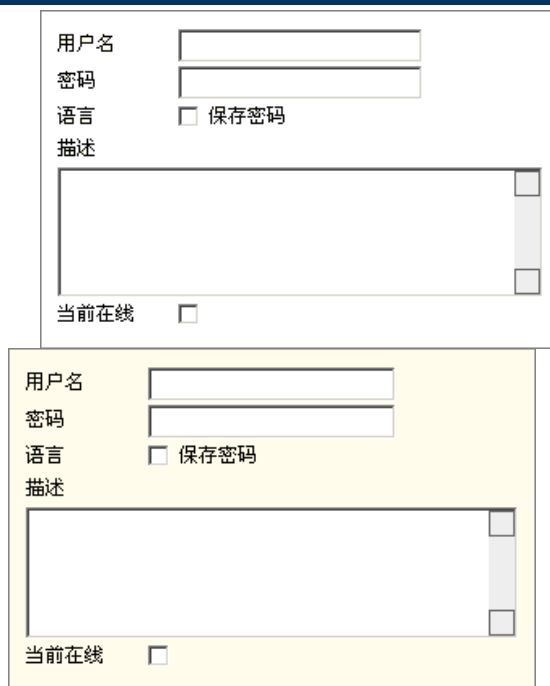


图 1:主组（左图）和从组（右图）。
使用主从组，为了

- 在全页面应用程序中定义一个 **area**。 注意：推荐使用从组（暗的）作为一个 **area** 的背景。然后你可以在其顶部放置一个突出显示的主组，或者用一个主组把部分 **area** 的文本包含起来。
- 突出显示（Highlight）一部分文本
- 将一系列的连贯元件组成一个组

Group Box

group box 样式（属性 `design = SAPCOLOR`）具有透明的主体背景，因为他的边框和 **header bar**，具有十分明显的外观。

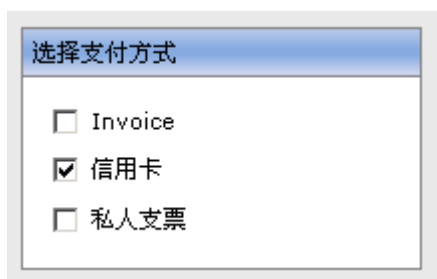


图 2: group box。

使用 **group box** 来聚集一系列联结使用的元件。不要在一个 **group box** 中嵌套任何组。避免使两个以上同类型的组之间过于接近，因为它们会创建一个类似 **grid** 的视觉效应，使用户难以判断。

样式相关属性

组的外观能由三种属性来决定：属性 **design** 用来选择组类型(可选值有 PRIMARYCOLOR, SAPCOLOR, SECONDARYBOX, SECONDARYBOXCOLOR, SECONDARYCOLOR)。属性 **width** 设置组的宽度，还有属性 **title** 设置标题文本。

细节参阅“组”页。

相关控件

标记栏 (tabStrip)，Table 视图，树状视图，文本视图（子组头部）

关于组的更多信息

界面编辑器的可编辑性

在界面编辑器中，以下组控件的属性可能被更改。

组	类型	IE5及以上版本	Netscape 4.7
字体	标题字体粗细	x	x
背景颜色	第一背景颜色	x	x
	第二背景颜色	x	x
	第三背景颜色	x	x
边框 布局	边缘宽度，样式和颜色	x	x
	包含Header Strip的组的标题填充 主体填充	x x	
	不含Header Strip的组的标题填充 主体文本填充	x	
	容器标题的背景高度	x	xx
	容器标题的背景颜色		xx
	容器主体的背景颜色		

11.9 图像

图像

图像控件支持 bitmap GIF 或 JPEG 格式，并且可以指定图像的宽度和高度。



图 1：在iView里包含一幅图片的例子。用户选择这张图片上的shuffler选项，可以显示该图片。

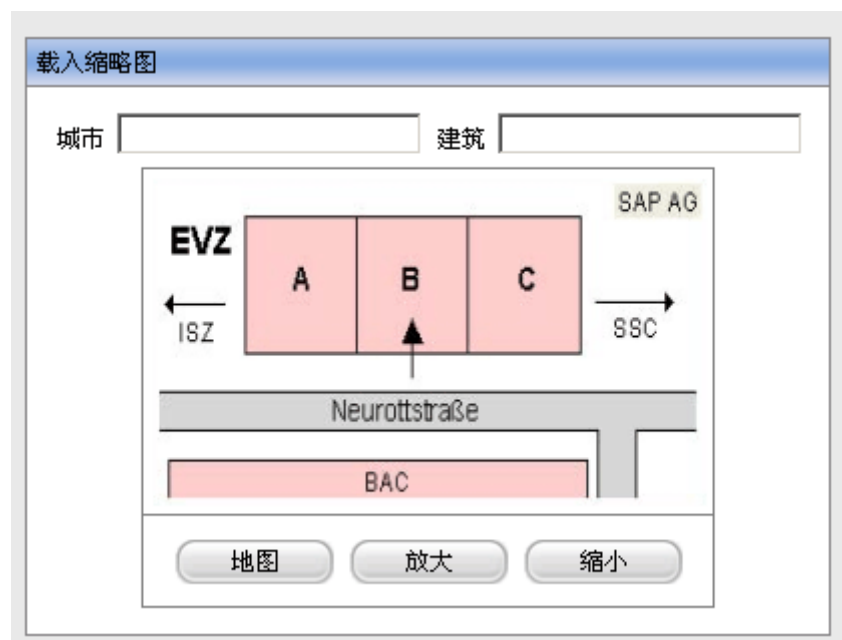


图 2：包含图片以及一组图形操作相关的按钮用法

图像控件支持 bitmap GIF 或 JPEG 格式，并能够指定图片的宽度和高度。

照片，图，图表和图解，地图，草图，动态图，和视频(影像)都能在页面中使用。与使用文字描述相比，合理地使用它可以节约大量的时间和屏幕空间，并且表达更多信息。

注意:尽管图标也是图像，除了用来显示状态信息，它们不被 iViews 使用。也就是说，在 buttons 或 tabs 上没有功能图标。

与图像的交互操作

- 如果希望能够从几个数据集中选出图和数据，或降低数据量，就要在图像上方设置一个过滤器或者 shuffler。(参阅图 1，上面)
- 在图像的下方左对齐的地方，放置表示状态信息的(例如缩放比例)和与图像操作功能相关的 buttons。(参阅上面图 2)
- 对整个 iView 进行操作的 buttons 放置在左下角；这些 buttons 可以与 table 相关的 buttons 放在同一行里。
- 如果存在重点 button，必须放置在相关 buttons 组中(有关图像或有关 iView)最左边的位置。在一个 iView 里一般只有一个重点 buttons。

图像说明

总是对图提供恰当的注释。根据图像的格式和 iView 或页面的布局，可以在图像的下面或者右边放置图像说明或其它文本。

使用图像的技巧

- 调整好图的位置，使它们的主要内容指向正文，而不是远离正文。
- 只显示相关部分的图像；尽量使用小图像，并且避免不相关且分散的元件。例如：如果你想要说明密歇根州的数据，改为只要使用密歇根州地图就可以，而不是整张美国地图。
- 使用高质量的图。例如：尽量使用图形设计软件绘图而不是手绘。
- 注意运用图像的正确格式：如果图像有复杂颜色或色调，应使用 JPEG 格式。GIF 格式用于色调单一并有粗体边界的图像，例如图解(diagrams)或卡通图像(少于 256 色)，通常，屏幕显示 GIF 图像比较清晰。有明显边界的图作为透明 GIF 可以在任

何背景下清晰显示。但是像在很多 logo 中使用的那样，将光滑的窗体和大规模文本保存成 GIF 格式时，如果你不知道窗体的背景颜色，可能会引起一些问题。

类型

动态图，视频或者影像



图 4: 动态图可以用来展示过程或仅仅看起来更有意思。
注意：目前图像控件只支持动态GIF格式。

草图

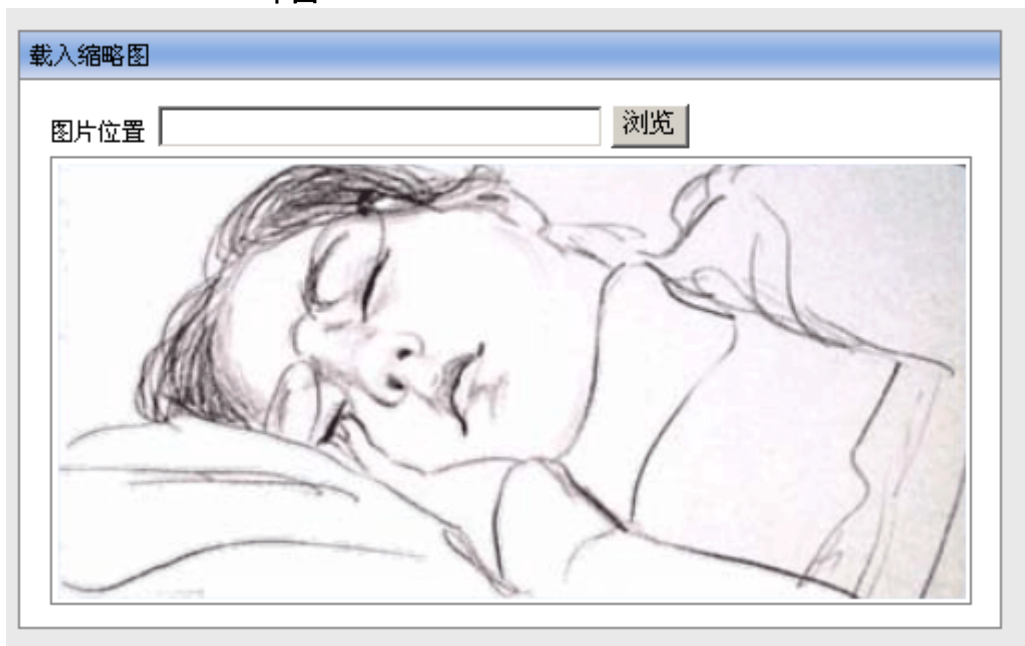


图 5: 草图是快速交流的有效方式，也可以作为一张图表的初级版本

设计相关属性

你可以为一幅图像的设置高度(height)和宽度(width)，也可以设置帮

助文本(tooltip)，tooltip 将在光标通过这幅图像或鼠标键被按下但未释放时展示出来。

用法 – 高度和宽度

注意：在浏览器内不要通过改变图像高度、宽度值来缩放图像。这可能会降低图像质量。

相关控件

Table 视图

关于图像的更多信息

浏览器兼容性 | 在格式编辑器里的可编辑性 | 可访问性 – 508 支持

浏览器兼容性

目前还未发现任何问题

在界面编辑器里的可编辑性

用户能个性化那些界面外部框架内使用并用来进行功能说明的界面图像 (iView 功能图像, table buttons, 等等), 通过界面编辑器实现起来十分容易。编辑器对那些作为界面内容的图提供了不可编辑的样式。

可访问性 – 508 支持

- 操作 (keyboard): 图像在缺省状态下不能够被插入到可访问层。
- 默认描述: 通过 HTMLB 解释引擎 提供。
- 应用-级描述: 如果需要的话, 通过使用 setTooltip 方法来设定附加的描述。 不要使用 setAlt 方法, 该方法用来设置替换的文本 (alt 属性)。

11.10 输入域

输入域

用法 | 类型 | 样式相关属性 | 相关控件

Figure 1 shows a collection of input fields within a light gray border. It includes:

- '用户名*' (Username*) with a red asterisk and a text input box.
- '重复用户名*' (Repeat Username*) with a red asterisk and a text input box.
- '性别' (Gender) with two radio buttons labeled '男' (Male) and '女' (Female).
- '日期' (Date) with a text input box and a clock icon on the right.
- '城市' (City) with a text input box.
- '地址' (Address) with a long text input box.

Input fields 在表单中被用于输入和显示数据。数据可以具有日期，整数或者字符串等多种类型。

input fields 可以包含不同的行为（例如：密码，只读，或必须的），或者不同状态（例如：正常和错误等）。

图 1: input fields集合的例子（有的要求必须输入，有的不要求必须输入）

用法

大多数情况下，input fields 有时会与标签控件绑定使用，有时也和附加的组件一起使用，例如描述或者按钮。

Figure 2 shows an input field with a dropdown menu on the left labeled '选项1' and three circular icons on the right: a magnifying glass, a print icon, and a question mark.

图 2: 和标签(左边)或附加组件一起使用的input fields

通常，标签被放置在 input fields 的左侧，而对描述紧跟着 input fields。也有一个例外：为了实现较紧凑的设计，标签放置在 input fields 的上方，这时标签可以使用小字体。

Figure 3 shows an input field with the label '项目名称' (Project Name) positioned directly above it.

图3:帮助文本可以被放置在input fields的左侧。

帮助文本

帮助文本是被在安置 input fields 后面的特殊描述—如果空间有限，可置于 input fields 左侧（图 4） 不要使用标签控件表示帮助文本，



图 4: 在input fields下面的帮助文本使用了小字体标签- 不要把标签控件用于帮助文本(在input field下方)

宽度对齐调整

通常 input fields 被组合, 用来形成一个输入数据的语义块, 例如地址数据或银行数据。那样的话, input fields 应该说明用户必须输入多少个字符。因此, 最好不要设置所有的区域为相同尺寸。这很重要, 因为 input fields 经常与其它输入类型一起使用, 例如 checkboxes 和 radio buttons 等。

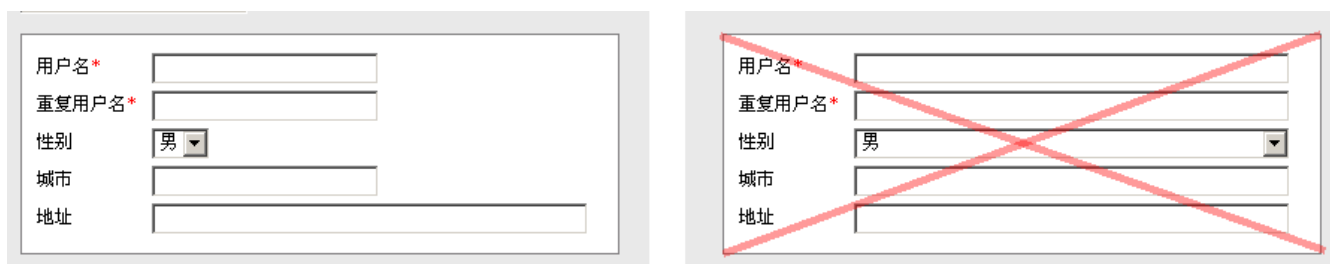


图 5:具有恰当宽度值的input fields分组的例子

使用 grid 布局控件来对齐 fields 和标签。Fields 和 labels 在 grid 内都向左对齐。标签和 input fields 之间的偏移量由最长的标签长度决定。

想要了解不同 input fields 类型和尺寸的用法, 请参阅下文。

类型

input fields 有两个尺寸: 标准尺寸和小尺寸。它们由属性 size 来设定: 标准尺寸(STANDARD 默认)或 (SMALL) 小尺寸。

用法 – 尺寸

通常, 只使用标准尺寸。如果实际屏幕大小受限制, 也可以使用小尺寸。不要在一个 field 组合中混合使用小尺寸和标准尺寸的 input fields。



图 6: 标准input fields(左)和小尺寸的input fields(右边)

样式相关属性

input fields 有许多属性，来表示不同的状态和行为，例如只读，错误，口令区和必填区。



图 7: input fields 的不同状态

可以通过设定 disabled, invalid, password, required 属性为 TRUE 来实现这些状态。

input fields 的数据类型，例如整数，字符串等，可以通过属性 type 来设置。欲了解可能的属性值和其他属性，请参阅页 input fields。

另外，可能存在伴随着帮助提示框的特殊 input fields。下面，我们以那些日期 input fields 作为例子(这是目前唯一支持帮助提示的 input fields – 更多的细节请参阅 input fields 页)

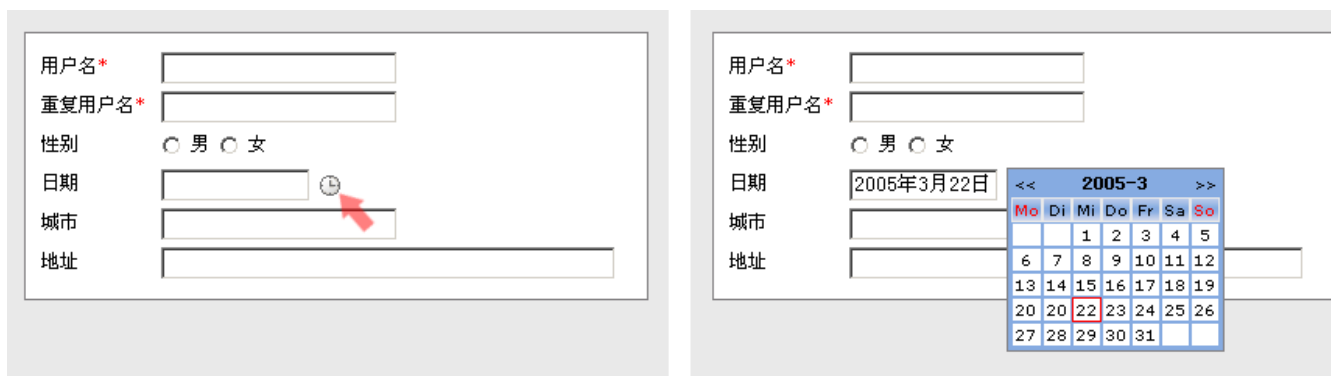


图 8:可选择日期的input fields

用法 – 必填的input fields

一些 input fields 要求在用户继续操作之前必需键入值（例如，在它们保存数据之前）。可以设置 required = TRUE 来使 input fields 成为必填的；它相应标签也必须设置成必填的。

用法 – 只读input fields

只读的 input fields(disabled = TRUE)是不允许用户输入数据的 input fields。

把用户已经填入数据的区域（例如，先前的页面或在先前的会话期间的信息）或由系统填入的区域，或当前用户不能更改的区域设置为只读区域。通常，允许用户输入的数据往往取决于显示在只读 input fields 中的受保护的数据。

用法 – 密码input fields

当用户必须输入口令时，例如，对于 login 对话框，使用口令 Input field(password = TRUE)。

用法 – 不合法的input fields (错误状态)

每当系统检测到用户提交的一个输入信息的错误时，为一个 input fields 设置错误状态(invalid = TRUE)。它取决于系统操作，必须同时显示一条错误提示信息，提示信息也可以在系统进行全面错误校验后达到“稳定”状态时出现。

关于不合法 input fields 的更多处理细节请参阅 错误处理。

相关控件

标签, Checkbox, Radio Button, Grid 布局, 流布局, 组

关于input fields的更多信息

浏览器兼容性 | 在编辑器里的可编辑性 | 可访问性 –
508 支持

浏览器兼容性

在任何 input fields 支持的浏览器中显示

在样式编辑器里的可编辑性

在样式编辑器里，可以修改下列 input fields 控件的属性值：

Group	Style	IE 5 and above	Netscape 4.7
Field Styles	Border Width, Style and Color	x	
	Padding	x	
Standard-Sized Field	Font Size	x	
	Height	x	x
Small-Sized Field	Small Font Size	x	
	Small Height	x	x
Invalid Field	Border for Invalid Input	x	
Required Field	Font Color of "Required" Indicator	x	x
Background	Background Color of Editable Fields	x	
	Background Color of Non-Editable Fields	x	

表格 1: input field控件可编辑的样式

可访问性 – 508 支持

input fields 必须与标签结合使用，用标签来表示该 input fields，这可以确保用户明白两个元件之间的关联，并且能依照标签找到相应的 fields。

操作: input fields 在缺省状态下能够被插入到访问层中。

- 默认描述: 是通过 HTMLB 解释引擎 提供。
- 应用-级 描述: 如果需要的话，通过 setTooltip 方法来设定一个附加的描述。
- 标签: 必须与一个标签控件相联系(使用方法 setLabelFor 来指定相应 input fields)

11.11 链接

链接

用法 | 类型 | 链接 vs. 按钮 | 状态 | 式样相关属性
| 相关的控件



图 1: 在content area内的链接的例子

用法

链接控件不像它看起来那样的简单。并非所有的链接都应该被平等处理。超文本链接在 navigation area, header area, content area 或者 button, Table 和应用内有不同的外观和行为。

为了美观和实用性,在导航 area 内的链接(不要把内容 area 和导航链接混淆在一起!)与在内容 area 中的链接有不同的外观。用户可以看到,这些链接是作为导航栏架构的一部分,而不仅仅用于获取详细信息或在不同上下文页面跳转,这是至关重要的。

这里描述的叫做"link"的 HTMLB 控件,在 content area 内指的是超文本链接。

布置

链接既可以作为大文本的一部分,(例如,一篇关于客户服务的文章可能包含一个 mySAP CRM 主页的链接)也可以单独出现(例如,关于 CRM 文章的一系列链接,在摘要的结尾处放置“更多”的链接来说明额外的信息等.)。

这些单独使用的链接应该组织在一起,但要同 buttons 区分开来,反之亦然。如果可能的话,功能和链接应该被分类并且以相同的

方式显示(无论是全部作为链接或全部作为 **buttons**)，如图 1 中的例子。 应该避免 **links** 和 **buttons** 在同组上下文内混合使用。

转换视图的链接经常出现在它所指内容的上方，就像图 2 所示。

书写规范

如果链接是大段文本的一部分或者是自动产生的(例如联系姓名，文档标题，等等)，不必为了某些链接做任何特殊的大写考虑。这同样也适用于 **buttons** 中的标题书写规范 (即第一个字母总是大写，所有重要的单词用大写字母表示，前置词和正文不要以大写字母表示)。功能，导航，**toggle** 和转换视图中的链接也一样。

类型

尽管从技术观点看只有一种链接类型，但我们必须能够辨别在 **content Area** 中的不同链接。这有利于建立链接的用法规则，并且帮助区分 **buttons** 和链接之间的差别。

注意

基于建立链接的不同目的，我们能建立 5 类不同的 **content area** 链接：

view switch, toggle, drill-down, function and navigation.

视图转换链接

视图转换链接 (**View switch links**) 在功能上相似于 **toggle** 链接，不同之处是前者总是可视的。视图转换链接是除标记栏 (**tabstrip**) 控件之外的另一种选择。视图转换链接与标记栏 (**tabstrip**) 相比较，优点是它占有较少的空间并且可以被垂直使用 (如果在应用程序中需要的话)。(例如移动应用程序中，设备的外形允许纵向空间大一些)。这类链接类似于导航链接。与导航链接相反，视图转换链接必须指向同样的内容，而且在所有页面内都要存在。当前选择的视图不应该还作为链接显示，而是以粗体文本显示(请参阅上面图 3)。视图转换链接应该被用一条垂直的线 (|) 与彼此隔开。

view switch links 要使用 **title case**. (请参阅大写规范)

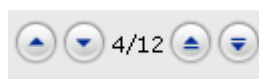


图 4: 包含toggle, 向下获取 (drill-down), 功能和导航链接的 iView

Toggle Links

Toggle 链接通常包含一对链接，但是只有一个可见的。通常使用

在对于当前数据有两种链接选择的时候，在图 4 中的 Toggle 链接显示" **Show Entire Feedback Text** "时，如果用户点击它，视图将显示全部反馈的内容，与此同时，链接的文本将转换成" **Show Only Feedback Preview** "。

一些 toggle links 的常见例子：

- Expand All / Collapse All
- Show Chart View / Show Table View
- Hide Help / Show Help
- 等等.

toggle 链接的文本要使用 **title case** 。（请参阅下面的书写规范）

向下获取链接（Drill-Down Links）

一个向下获取链接允许用户看见更详细或更具体的信息。例如，在通讯录的目录中，联系姓名是向下获取链接，它允许用户访问那个人的详细信息。在邮件收件箱里，每条消息的标题将自动作为一个链接，链接到消息的具体内容。

一些常见的向下获取链接：

- 联系姓名
- 用户姓名
- 文档标题
- 信息标题
- 报告标题
- 收入

这些链接一般都会被自动生成，开发者不应该试图改变它的书写格式。（参阅图 2.）

功能链接

功能链接允许用户执行一个动作。

尽管常规的 **button** 都会提供功能性的操作（更多信息参阅 **Links vs. Buttons**），但在某些情况下，使用链接更合适，或者某些情况下，功能和一个指向其它页面的链接之间的区别是微乎其微的。

例如，当用户想要预订一个对象(例如一个文件夹)时，链接也许只是一个将用户引导到新界面的链接，在这个界面中，用户填入一些信息并把信息提交给服务器。这个的预定行为，可以被认为是执行了一个功能，并且有时候使用 **button** 比使用链接，更能体现这一动作，但反之亦然，这取决于上下文。

有时你可以有一个 **ITEM** 的集合，例如一系列的文档。有些操作你可以对所有的文档一次执行，然而有时，一次仅对一个文档操作会更好。如果操作执行后会请求一个新页面或获得额外的信息(细节，反馈，编辑，回答，等等)，或者只对一个 **ITEM** 处理，用 **Link** 会比用 **button** 更适合。

把图 1 中的 **iView** 作为一个例子，我们可以看到用户可以选择一批文档并使用 **checkbox** 和 **button** 一次性的进行批准或拒绝操作。但是，如果用户想要看这批文档的细节，让用户选择紧挨着每份文档的链接来进行操作是更为合适的。否则，用户将会看到一批文档的所有细节，他们会为不知道那些细节是属于哪个文档而感到困惑。

如果你的应用有一系列的 **item**，并且每一项都要求实现它自己的功能(请参阅图 1 和 4)，在这种情况下最好是使用 **link** 而非 **button**。这主要是为了美化外观，但也考虑了用途的因素。应用中有一堆 **button** 会使它看起来非常繁杂和难以理解。

一些常见的功能链接的例子：

- 详细资料
- 回应
- 增加
- 预订
- 答复
- 编辑
- 等等.

功能链接的文本要使用 **title case**。(请参阅书写格式)。

导航链接

很多时候，**iView** 或应用中也可能有独立于界面主导航栏之外的导航栏。终端用户甚至意识不到这些链接也起到" 导航"的作用。 导航链接允许用户在数据集或操作过程中前后移动。 有时候 **drill-down** 链接和导航链接之间的差别很难界定，例如一个"**more...**" 链接。

一些常见的导航链接是：

- 更多...
- 下一个 or 下一个 > ,
- 上一个 or < 上一个 ,
- 向后 "<"
- 向前
- ">>" 回溯到开始处
- "<<" 回溯到结束处
- ">>" arrows in text form (如上图 4 中所示)
- 数值 作为导航链接(如上图 4 中所示)

导航链接的文本要使用 title case (请参阅书写格式 (Capitalization))

链接 vs. Buttons

通常，使用链接表明了要跳转到另一 HTML 页面，或者切换到当前信息的不同视图或链接到更详细的细节信息。链接通常在应用的上下文内出现(trees, tables or text)。

通常，使用 buttons 表明了一个功能可被执行(save, print, close, delete 等等)或一个过程可以开始(预订等等)。Button 一般在一组区域的左下方出现，来表明一个功能能够在被选定的 item 上执行(如果 checkboxe 也出现)或它适用于整个界面。(请参阅指南的 Buttons for further information 部分)

更多在哪里链接可被用来指示一个功能的信息，请参阅上面的功能链接。

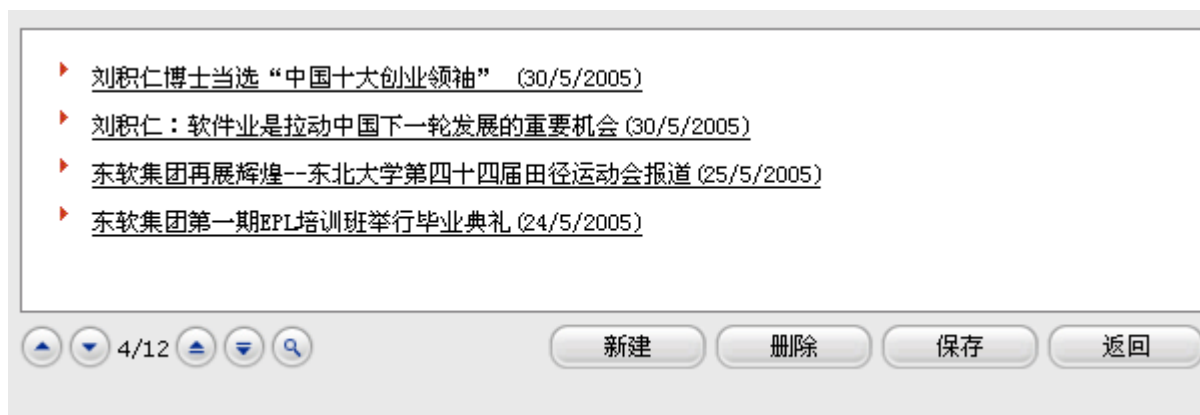


图 5: iView中包含链接和button的例子。

状态

链接控件有 4 种状态：链接(即未访问的，普通状态)，悬浮，已访问的，活跃的。

样式相关属性

reference 属性允许把 URL 指派给一个链接，而 **target** 属性允许设置一个链接目标。后者遵照 HTML 惯例来指定链接目标。**text** 属性设置链接文本，**tooltip** 属性设置链接的 tooltip 文本。

更多细节请参阅 [链接页](#)

相关控件

Button

有关链接的更多的信息

[浏览器兼容性](#) | [界面编辑器的可编辑性](#) | [可访问性 – 508 支持](#)

浏览器兼容性

Netscape 不能够识别悬浮(hover)状态.

样式编辑器中的可编辑性

在界面编辑器中可以设置下面的界面风格：

不可访问的 ("链接"), 活动的 ("活动的"), 已访问的 ("已访问的") and Mouseover ("hover ")的字体颜色

不可访问的("链接"),活动的("活动的"),已访问的("已访问的")，鼠标划过（划过），的文本式样

关于常用界面风格概述 请参阅 Customer Branding and Style Editor 的 HTMLB 控件和界面编辑器 部分。

可访问性 – 508 支持

操作 (Keyboard)：链接以缺省的方式插入到访问层。

默认 描述: 是通过 HTMLB 解释引擎 提供。

应用-级描述: 如果需要的话，使用 **setTooltip** 方法设置一个附加的描述。如果用户需要更多的具体的信息或说明的话，则需要附加的描述。通常，如果链接引入的交互操作不能被初级用户识别时，就必须扩展描述。例如，如果链接将打开一个新的窗口，必须添加描述。

11.12 列表框

列表框

用法 | 式样相关属性 | 相关控件

列表框是一个显示条目列表，以便用户可以选择的框架。如果条目数量超出了列表框的大小，会激活一个垂直的滚动条。列表框是只读的。



Figure 1: 列表框控件的例子

用法

一个列表框提供一系列的选项，以便用户可从中选择其一。如果条目的数量超过控件的尺寸，会激活一个垂直的滚动条。在列表框里的一个条目被称为列表框条目。列表框是只读型的。

注意：列表框控件不会自动显示一个描述标签。可以使用标签控件为 **list box** 增加描述。考虑一下，如果要突出显示标签，你要怎样改变文本的属性，例如，使它变为粗字体（参阅图 1）。

选择适当的Selection Control

列表框在功能上相似于下拉列表框 – 它们都提供了用户能从中选择的条目列表，即，两个都是单选列表。

参见 **Forms** –应用不同类型的列表来指导选择合适的选择控件

注意：对于较小的条目数量(2-6 个)，并且全部条目必须对于用户可见，可以使用 **radio button**。

式样相关属性

你可以设置列表框显示的行数 (属性 **size**)，和它的宽度(属性 **width**)，还有它是否可用（**disabled** 属性的布尔值）。

参阅 列表框细节页

相关控件

下拉列表框，条目列表，Radio Button，Table 视图，树状视图

关于列表框的更多信息

浏览器兼容性 | 样式编辑器中的可编辑性 | 可访问性-508 支持

浏览器兼容性

将要被提供

样式编辑器中的可编辑性

列表框自身作为标准浏览器控件显示，样式编辑器可以对相应的标签进行更改。

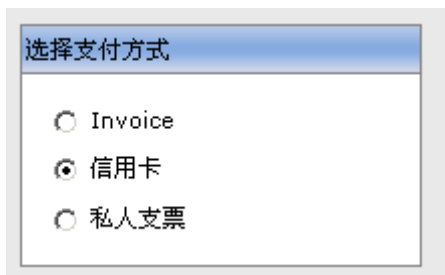
可访问性-508 支持

列表框必须与指向指定列表框的标签联合使用。这可以保证用户明白两个组件之间的关系，并且能够理解下拉列表的正确标签。

- 操作（Keyboard）：列表框默认插入访问层
- 默认描述：由 HTMLB 解释引擎提供
- 应用-级描述：使用“setTooltip”方法设置一个附加的描述
- 标签：必须同一个标签控件绑定使用（使用“setLabelFor”方法识别相应的列表框）

11.13 单选按钮

用法 | 式样相关属性 | 相关控件



Radio button 可让用户从一系列选项中进行单选

Figure 1: 一个radio button组

用法

Radio button 让用户从一系列选项中进行单选。它们总是在一组 radio button(至少两个)里出现。因此，你只能在 radio button 组内定义 radio buttons，不能作为单个元件

选择一个选项并自动取消之前的选择。通常，初始化形态中，会有一个 radio button 以被选中的形式出现在整个 radio buttons 组中。但网络应用中会出现一些特例，有时候，初始状态下 radio button 组中可以没有被选中的 radio button。

注意

radio button 组内的水平间距一般不能被更改。如果你需要设置不同的间距，如果可能的话，可以结合单个 radio button 和 Grid 布局控件来实现。

排列和样式选项

关于 radio button 的排列和式样选择的细节参阅 Forms - 使用 Radio Button。

样式相关属性

Radio button 有 disabled 属性。若暂时不允许用户改变 Radio buttons 状态，设置 disabled 属性值为 TRUE。text 属性为 radio button 设置标签文本描述。

浏览器兼容性

radio button 可以在任何支持它的浏览器中显示

样式编辑器里的可编辑性

`radio button` 是浏览器的标准控件。样式编辑器可对其标签进行更改。

单选按钮组

在样式编辑器里不支持对 `radio button` 组的编辑。

可访问性 – 508 支持

如果 `radio button` 要与它左侧的标签一起使用，`radio buttons` 必须与对应的标签控件组合使用。这保证用户知道两个组件之间的关系，并且能理解 `radio button` 相应得标签。

- 操作（Keyboard）：Radio button 缺省插入访问层
- 缺省描述：由 HTMLB 解释引擎提供.
- 应用-级描述：如果需要的话使用 `setTooltip` 方法添加附加的描述.
- 标签：必须同其左侧的标签一起使用(`setLabelFor` 方法识别相应 `radio button` 或者 `radio button` 组)。

11.14 表视图

表视图
用法 | 类型 | 样式相关属性 | 相关控件

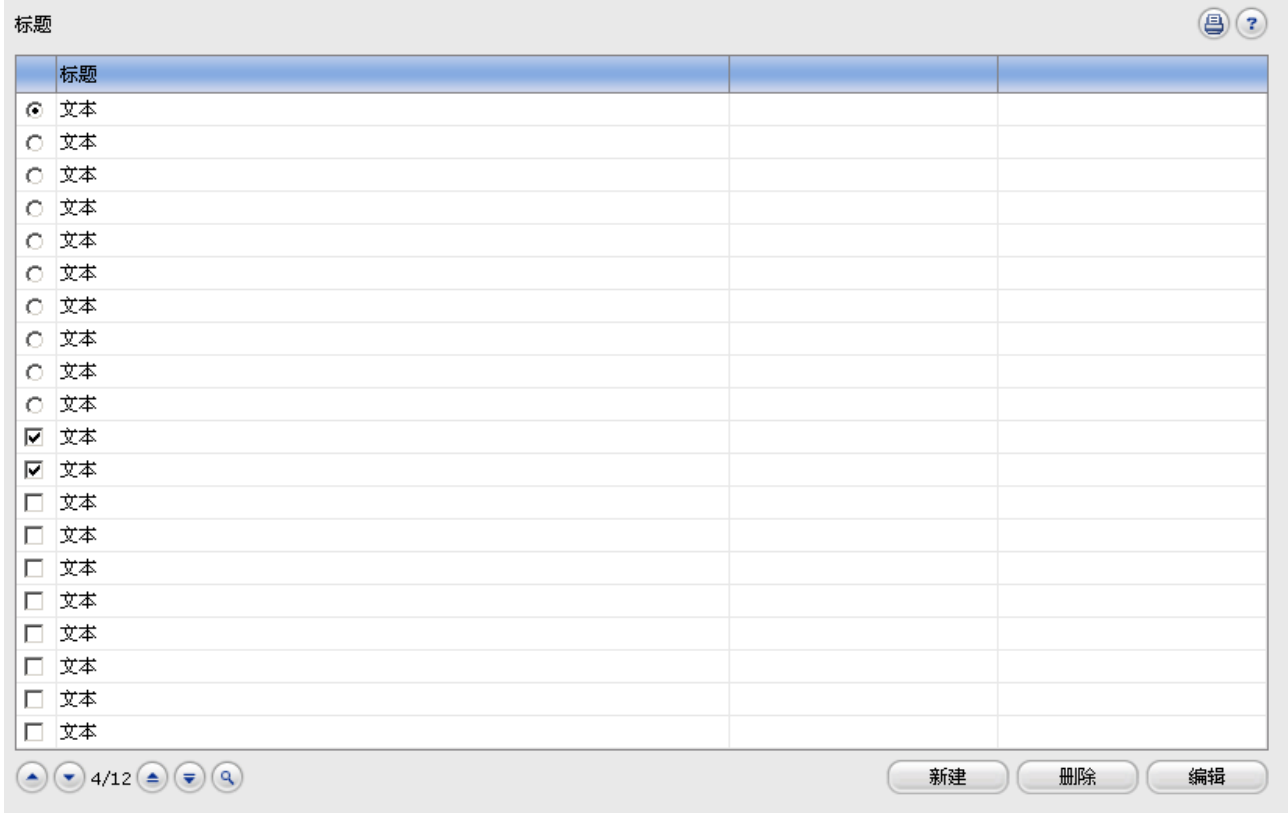


图 1:含有不同列类型和错误输入域的table视图

table 视图在行与列中，可以排列数据-文本，图像，链接以及其它 table 等等（以 tabular 的形式）。Table 视图的行分为 header、body、footer 三部分。Table 视图支持导航按钮来在 Table 中滚动。另外，Table 视图提供了单选和多选行机制。

用法

Table views 主要作为数据表格，用来显示数值或者非数值的 tabular 数据。Table 视图 views 可以是只读，也可以用于数据输入，根据 Table 视图的用途，可以选择不同的外观和行为。

一般使用技巧

Table 是相对复杂的界面元件，开发人员可以添加很多的信息。但在保证合理行和列数量的前提下，应尽量减小 Table。

对于长 Table，考虑 shuffler 一类的过滤方法，它可以使 table 只显示几行，用户基本不需要滚动，或仅需小幅的滚动。

此外，也可以考虑使用其它的表示方式，例如图表或者图型（charts or graphs） 有时它们可以更迅速地展示相关信息。

外观

Table 视图可由 3 种可选择的外观形式：

- Grid and background -可以在行之间交替使用不同颜色或者统一背景颜色。
- Transparent,在页面内统一 grid 的背景风格

Grid and uniform background: 使用这种外观对于输入型 Table 和显示数字的 Table 更合适，统一颜色背景也适用于窄的或中等宽度的 table。

Stripe pattern background: 使用 stripe pattern 更加适合数据输入型 Table 和宽 Table 显示.(可能是对每一行都用不同颜色交替显示)

Transparent background: 使用透明背景适合于无数字的 Table,即那些只显示文本或者图像的 Table 使用。

Table标题和Table的组成部分

一个 table 视图由三个主要部份组成，标题（header）行，table 视图主体，还有页脚（footer）行。

- Table 视图的标题行包含 Table 视图的标题。如果 Table 视图在其它地方没有被描述（例如被组或者 tray title 描述），那么它就应该有一个标题。
- Table 视图主体包含真正数据
- Table 视图 footer 位于最底行，它包含一个靠左的滚动 buttons 和一个可选使用的文本区。例如：footer 文本可以提供翻页信息。

Header 行和 footer 行都可以被隐藏，注意，隐藏了 footer 的同时也就隐藏了滚动 buttons。

行和列的标题Headers

行和列的标题（headers）描述了行和列的数据。典型地，一个 table 视图只有一些列标题，它们描述了数据行中的各数据项的不同属性。而行标题可用在有行和列标题的类似矩阵的 table 中。

单元格内容

Table 视图的单元格可以包含文本，图像，图标，链接，button，输入域和下拉列表框。对于独立输入域，Table 中的输入域也有不同的属性，例如必填，只读，或者错误状态。

行的选择

如果需要，table 视图可以有两种选择模式

- 单选模式（在第一列中放置 radio button）
- 多选模式（在第一列中放置 checkboxes）

这两种模式通过 `selectionMode` 属性的 `SINGLESELECT`，`MULTISELECT` 值来设定。Table 视图的缺省模式(`selectionMode = NONE`)下，用户不能选择任何行。如果不需要用户选择行或某些项，可以使用这种模式。如果用户只需选一行，则使用单选模式。如果用户需要平行的选择若干个选项或条目，则使用多选模式

Scrolling

Table 视图为 Scroll 功能提供 6 种 button: First page, Page up, Line up, Line down, Page down, 和 Last page。（首页，上页，上一行，下一行，下页，尾页）

如果 footer 被隐藏起来，scroll button 是看不见的，注意：目前 Table 视图中不支持 Scrollbars。

技术信息：

只有属性 `selectionMode` 设为 `NONE` 或者 `SINGLESELECT` 时，Line up/down 按钮（上一行，下一行）才能显示出来

Table视图尺寸

关于表格视图尺寸的建议：

垂直尺寸

在垂直方向上，Table 视图应适合容纳它的窗体或者 iView。尽管 Table 视图可以通过 button 来滚动观看，一般不要使用窗体的滚动条（scrollbar）。

应该尽可能让 Table 视图的高度与包含容器的高度一样。Table 视图越大，就可以减少滚动的需要（通过按钮滚动是很麻烦的）。

Table 视图应该有至少 3 行可见 - 5 行更好。

水平尺寸

水平方向上，Table 视图应该适合包含它的窗体或者 iView。尽量避免水平滚动。

矩阵Table

Table 表格应该至少有 2x2 的数据单元格。

初始尺寸和外观

空的 Table 不会在 Views 中显示。在这种情况下，不要出现没有 table 数据或像 "No entries found" 这样的信息。

例外： 这个规则的特例： 用户能立即输入数据的 Table 应该以预设的大小出现，并带有若干空行。

注意： 这种情况下不要使用透明样式。

单元格式样

Style	Value
STANDARD	Text
NEGATIVE	-2.000.000,35
POSITIVE	2.000.000,35
TOTAL	2.000.000,35
SUBTOTAL	0
SUBTOTAL_LIGHT	0
BADVALUE_DARK	-2.000.000,35
BADVALUE_MEDIUM	-2.000,35
BADVALUE_LIGHT	-1,35
CRITICALVALUE_DARK	-0,35
CRITICALVALUE_MEDIUM	-0,15
CRITICALVALUE_LIGHT	-0,01
GOODVALUE_DARK	2.000.000,35
POSITIVE	2.000,35
GOODVALUE_LIGHT	20,35
GROUP_HIGHLIGHTED	Use to emphasize group data, never together with critical colors
GROUP_HIGHLIGHTED_LIGHT	Use to emphasize group level 2 data, never together with critical colors
KEY_MEDIUM	Use for Key Values Unique Identifiers
GROUP_LEVEL1	Use to group like a tree level 1
GROUP_LEVEL2	Use to group like a tree level 2
GROUP_LEVEL3	Use to group like a tree level 3
MARKED	Use to show that a ROW is selected

图 3: 显示不同单元格式样的Table - 点击图片看更大视图

注意： " GROUP_HIGHLIGHTED "颜色不应该被与 "CRITICALVALUE"颜色联合使用。

单元格内容类型

典型地，table 单元格包含数字的文本，或文字数字混合的文本。不过，还存在更多的单元格类型：

- Text: 文本单元格 – 文本不能够被编辑
- Image: 单元格显示图标或图像
- Link: 单元格包含一个链接（供参考）

- Button: 单元格包含一个 button
- Input: 单元格可以被编辑
- User: 单元格包含一个下拉列表框

Different TableColumnType settings					
Text	Image	Link	Button	Input	User
Hello World		Apple Computer Inc.	Click Me	Enter something	Midnight ▼
Good morning		IBM	Click Me	Enter something	Midnight ▼
Good morning		IBM	Click Me	Enter something	Midnight ▼

图 4: 显示不同单元格类型的 table

样式相关属性

决定 table 的外观和行为地多个属性:

- Header and Footer: 标题文本(header Text)可以被定义, 同 footer 一样, 也可以被隐藏(Boolean 属性 headerVisible, footerVisible)。 注意: 如果表格具有滚动条, footer 必须显示出来。(也就是说, 在屏幕可显视范围以外, Table 还含有更多行)。
- Size, Number of Lines, Initial Appearance: 这些属性决定了表格视图的宽度(width), 可视行的数量(visibleRowCount), 第一个可视行(firstVisibleRow), 以及空行的初始外观(布尔属性 fillUpEmptyRows)。
- Selection Mode: 定义了 Table 条目被如何选择: 单选, 多选, 或不选(属性 selectionMode, 值 SINGLESELECT, MULTISELECT 或 NONE)。 注意: 如果 footer 被设置成可见的, Selection Mode 也影响 scroll button 的显示。

更多细节请参阅 Table 视图部分

相关控件

Tree View, Item List, List Box

Style	Value
STANDARD	Text
NEGATIVE	-2.000.000,35
POSITIVE	2.000.000,35
TOTAL	2.000.000,35
SUBTOTAL	0
SUBTOTAL_LIGHT	0
BADVALUE_DARK	-2.000.000,35
BADVALUE_MEDIUM	-2.000,35
BADVALUE_LIGHT	-1,35
CRITICALVALUE_DARK	-0,35
CRITICALVALUE_MEDIUM	-0,15
CRITICALVALUE_LIGHT	-0,01
GOODVALUE_DARK	2.000.000,35
POSITIVE	2.000,35
GOODVALUE_LIGHT	20,35
GROUP_HIGHLIGHTED	Use to emphasize group data, never together with critical colors
GROUP_HIGHLIGHTED_LIGHT	Use to emphasize group level 2 data, never together with critical colors
KEY_MEDIUM	Use for Key Values Unique Identifiers
GROUP_LEVEL1	Use to group like a tree level 1
GROUP_LEVEL2	Use to group like a tree level 2
GROUP_LEVEL3	Use to group like a tree level 3
MARKED	Use to show that a ROW is selected

关于表格视图的更多信息

浏览器兼容性|样式编辑器的可编辑性 | 可访问性 – 508
支持

浏览器兼容性

将被提供

在样式编辑器的可编辑性

在界面编辑器中，可以修改下面 table 视图控件的属性：

Group	Style	IE 5 and above	Netscape 4.7
Table Styles	Background Color of Standard Table Cell	x	x
	Background Color of Alternating Table Cell	x	x
	Grid Color	x	
	Cell Height	x	
	Cell Padding	x	
	Background Color of Selected Cell	x	x
	Background Color 1 of Grouping Cell	x	x
	Background Color 2 of Grouping Cell	x	x
	Background Color 3 of Grouping Cell	x	x
Table Icons	Background Position	x	
	Height	x	
	Width	x	
	Padding	x	
	URL to "Top" Icon	x	x
	URL to Inactive "Top" Icon	x	x
	URL to "Page Up" Icon	x	x

	URL to Inactive "Page Up" Icon	X	X
	URL to "Up" Icon	X	X
	URL to Inactive "Up" Icon	X	X
	URL to "Down" Icon	X	X
	URL to Inactive "Down" Icon	X	X
	URL to "Page Down" Icon	X	X
	 URL to Inactive "Page Down" Icon	X	X
	URL to "Bottom" Icon	X	X
	URL to Inactive "Bottom" Icon	X	X
Container	Font Color of Container Title	X	X
	Background Color of Container Body	X	X

表格 1: 表格视图控件的可设定的风格

关于常用界面风格的概述 请参阅HTMLB Controls and Style Editor in Customer Branding and Style Editor部分。

可访问性 - 508 支持

- 操作 (keyboard): table 视图并不是缺省插入可访问性层次的。
- 缺省描述: 由 HTMLB 解释引擎提供 table 中的标题, 导航 button 和特定的元件。
- 应用级描述: 调用 SetHeaderText 方法设置 table 的标题; 调用 setSummary 方法设置摘要正文(应显示一个工具提示). 注意: 没有 setTooltip 方法来设置 table 视图, 而要使用 setSummary 代替。Result

标题

?

	标题		
<input checked="" type="radio"/>	文本		
<input type="radio"/>	文本		
<input type="radio"/>	文本		
<input type="radio"/>	文本		
<input type="radio"/>	文本		
<input type="radio"/>	文本		
<input type="radio"/>	文本		
<input type="radio"/>	文本		
<input type="radio"/>	文本		
<input type="radio"/>	文本		
<input type="radio"/>	文本		
<input checked="" type="checkbox"/>	文本		
<input checked="" type="checkbox"/>	文本		
<input type="checkbox"/>	文本		
<input type="checkbox"/>	文本		
<input type="checkbox"/>	文本		
<input type="checkbox"/>	文本		
<input type="checkbox"/>	文本		
<input type="checkbox"/>	文本		
<input type="checkbox"/>	文本		
<input type="checkbox"/>	文本		

4/12

新建

删除

编辑

11.15 标签栏

标签栏

图 1: 使用单个（标签）tab 和（标签卡）tab card 实现的标签栏（tabstrip）控件例子

标签栏（tabstrip）是一个容器，它允许用户通过点击 tabs 在不同的视图之间切换。各视图共享屏幕上的同一区域。用户可以点击 tab 访问一个特定的视图。

用法

优点

标签栏（tabstrip）可以让用户一次看到所有可选择视图的标签。这样用户使用固定的标签栏，就能容易的在不同视图间浏览。

标签栏（tabstrip）对于呈现多重视图是一个理想的选择，尤其是当视图间的内容差异很大，用不同的形式来显示会带来杂乱的界面时。

缺点

标签栏（tabstrip）的缺点是与其它视图切换工具相比要占用大量的空间（例如 radio button 或下拉菜单 shuffler）。相对于标签栏

（tabstrip）来说占有较少空间的其它视图和其它选择的信息，具体请参阅相关控件。（对于标签栏来说，视图和其它的元件会占用更少的空间）

标签栏（tabstrip）的另一个缺点是视图的数量受 tab 空间大小的限制。

推荐（功能）

标签栏（tabstrip）可以容纳动态信息，因此用户可以迅速得到在各视图内重要的数据，事件或变化的概况。

（标签卡）tab card 内可以容纳 table 和 group boxes。

不推荐

避免在 tab 标签（tab labels）中使用过长的名字以及使用太多的 tab，因为这会导致控件过宽，也会产生例如横向滚动条或过宽 iView 的问题。

tab 不可以包含图标。

标签栏（tabstrip）提供给用户可按任意顺序访问的视图，如果不在这种场景下，则避免使用标签栏（tabstrip）。

尽管标签卡（tab cards）空间有限，仍不应使用滚动条。

标签栏（tabstrip）不可以互相嵌套。

一般使用技巧

仅当其他可选工具可能会导致使用户困惑的不稳定界面时，使用标签栏（tabstrip）来选择视图： 标签栏（tabstrip） 看上去相当大，会占据很多屏幕空间。另外标签栏（tabstrip）应该只用于没有指定步骤顺序的任务，因为它们可以被按任意顺序访问。

式样相关属性

标签栏（tabstrip）的外观和动作受多种属性影响。

高度和宽度：属性 bodyHeight 设置标签栏（tabstrip）面板的垂直高度，属性 width 设置标签栏（tabstrip）的宽度。

水平和垂直 tab 队列（Horizontal and Vertical Alignment of Tabs）：使用属性 horizontalAlignment（值为 CENTER, CHAR, JUSTIFY, LEFT, RIGHT）和 verticalAlignment（值为 BASELINE, BOTTOM, MIDDLE, TOP）排列 tab。

已选 tab：使用属性 selection 选择一个 tab。

Tooltip 文本（Tooltip Text）：使用属性 tooltip 为标签栏（tabstrip）总体上设置 tooltip 文本。

另外对每个视图（或 tabstrip item），下面有几个属性可单独设置：

高度和宽度：使用属性 height 设置标签栏（tabstrip）视图的垂直高度，属性 width 设置视图的宽度。

tab 文本：使用属性 title 设置 tab 标签。

Tooltip 文本: 使用属性 tooltip 为 标签栏 (tabstrip) 视图设置 tooltip text。

相关细节请参阅 标签栏 (tabstrip) 。

相关控件

radio button, 下拉菜单 (shufflers)

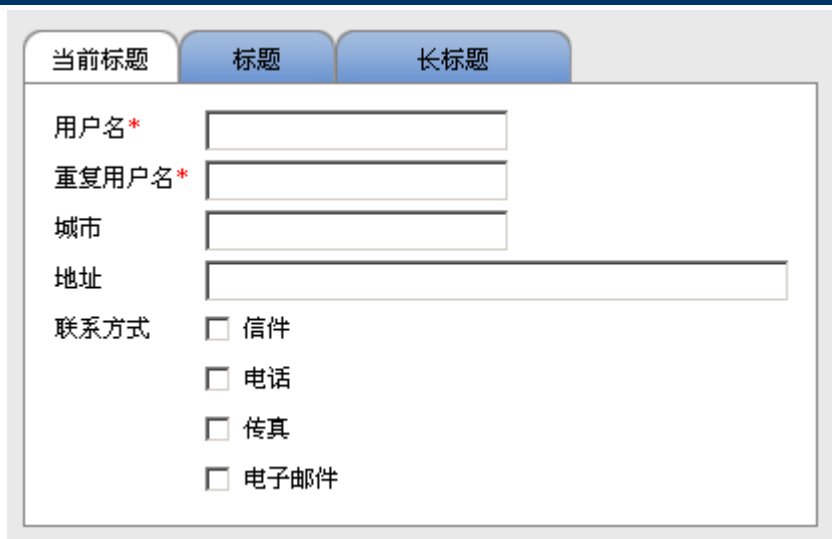
关于标签栏 (tabstrip) 的更多信息

浏览器兼容性

Netscape Navigator 的某些版本不能显示标准标签栏 (tabstrip) 控件视图的某些细微差别。

图 1: 标准标签栏 (tabstrip) 的例子

该标签栏 (tabstrip) 视觉上比标准标签栏 (tabstrip) (图 1) 简单的多 — tab 和 tab card 之间没有边框, 不同 tab 之间有空隙, 并且活动 tab 的高度与非活动 tab 的高度相同。



A web form with a tabbed interface. The tabs are labeled '当前标题' (Current Title), '标题' (Title), and '长标题' (Long Title). The '当前标题' tab is selected. The form contains the following fields and controls:

- 用户名* (Username*) with a text input field.
- 重复用户名* (Repeat Username*) with a text input field.
- 城市 (City) with a text input field.
- 地址 (Address) with a text input field.
- 联系方式 (Contact Information) with four checkboxes:
 - ☐ 信件 (Letter)
 - ☐ 电话 (Phone)
 - ☐ 传真 (Fax)
 - ☐ 电子邮件 (Email)



A web form with a tabbed interface, identical to the one above. The tabs are labeled '当前标题' (Current Title), '标题' (Title), and '长标题' (Long Title). The '当前标题' tab is selected. The form contains the following fields and controls:

- 用户名* (Username*) with a text input field.
- 重复用户名* (Repeat Username*) with a text input field.
- 城市 (City) with a text input field.
- 地址 (Address) with a text input field.
- 联系方式 (Contact Information) with four checkboxes:
 - ☐ 信件 (Letter)
 - ☐ 电话 (Phone)
 - ☐ 传真 (Fax)
 - ☐ 电子邮件 (Email)

Tabstrip Items

Tabstrip items 不能存储在 web 客户端。应用程序必须管理标签栏 (tabstrip) items。因此改变 tab 总能触发 tabSelectionChange 事件。推荐做法是至少要声明一个方法，当应用程序在一个 Netscape Navigator 4 Web 客户端打开时没有异常抛出。

界面编辑器的可编辑性

在界面编辑器中，可以改变所有的背景和边框颜色，以及 padding 和所有文本属性。下面是可改变的式样的列表：

Group	Style	IE 5 and above	Netscape 4.7
Tabstrip Styles	Background Color of Selected Tab	x	x
	Background Color of Inactive Tab	x	
	Left Border of Inactive Tabs	x	
	Right Border of Inactive Tabs	x	
	Top Border of Inactive Tabs	x	
	Tab Padding	x	
	Tabstrip Border Color		x
	Tab Height		x
Container	Container Border	x	
	Top Border of Container	x	
	Right Border of Container	x	
	Left Border of Container	x	

表 1: 标签栏 (tabstrip) 控件的可编辑式样

常用式样请参阅 HTMLB Controls and Style Editor in Customer Branding and Style Editor.

可访问性 - 508 支持

- 操作 (keyboard): 标签栏 (tabstrip) 的每个 tab 缺省情况下插入可访问性层次。
- 缺省描述: 由 HTMLB 解释引擎提供给每个 tab。
- 应用级描述: 如果需要, 使用 setTooltip 方法为每个 tab 设置附加描述。

11.16 文本编辑控件

文本编辑控件

Figure 1 shows a user interface example. It includes labels for '用户名' (Username), '密码' (Password), '语言' (Language), and '描述' (Description). The '描述' label is positioned above a large text area labeled '文本编辑' (Text Editing). To the right of the '语言' label is a checkbox labeled '保存密码' (Save Password). At the bottom left, there is a checkbox labeled '当前在线' (Currently Online).

图 1: iView 中的文本编辑控件例子

文本编辑控件为多行文本编辑提供工作区

用法

使用文本编辑控件允许用户编辑多行文本。

除非使用 HTML 命令，文本被限制到单个字体、大小和式样。文本编辑控件具有框架，框架大小由属性 `rows` 和 `cols` 定义。垂直滚动条（vertical scrollbar）一直存在，当文本行数超过可视行数时滚动条可用。

排列布局

有两种可能的方法来排列文本编辑控件：

- 在一组描述性 label 之下，在文本编辑控件之上 (图 1)。
- 一行中有其它域在同一域组中，label 放在文本编辑控件左侧 (图 2)。

如果文本编辑作为主要信息，请把它置于 `field` 组之下，而置于它上面的 `fields` 只提供在文本编辑控件中内容的描述信息。例如：当发送一个问题到服务中心时的问题描述。如果描述信息只是其他信息中的一条时，把文本编辑 `field` 放在 `field` 组中。文本编辑控件被用于一个形式自由的多行 `input field`。例如：一个消费者在网上商店注册时输入他（她）的各种参数。

式样相关属性

文本编辑控件可通过一些属性来改变：

- 设置文本区和 tooltip 文本 (tooltip text) (属性 text 和 tooltip)。
- 设置行数和列数(属性 rows 和 cols)。
- 确定折行 (属性 wrapping, 值为 HARD, SOFT 和 OFF)。

相关控件

文本视图

关于文本编辑的更多信息

浏览器兼容性

在 Netscape 4.7 中边框以 3D 的效果显示；背景总为白色；disabled 与 enabled 外观相似。

界面编辑器可编辑性

在界面编辑器中，可以修改下列文本编辑控件的属性：

Group	Style	IE 5 and above	Netscape 4.7
Text Edit Styles	Padding	x	
Container	Container Border	x	

表 1: 文本编辑控件的可编辑式样

常用式样请参阅 HTMLB Controls and Style Editor in Customer Branding and Style Editor.

可访问性 - 508 支持

文本编辑控件必须与指向已指定的文本编辑控件的 label 元件结合使用。这可确保用户能清楚意识到这两个元件之间的关系，并且能了解与文本编辑控件相对应的正确的 label。

- 操作 (keyboard): 文本编辑控件缺省插入到可访问性层次中。
- 缺省描述: 由 HTMLB 解释引擎提供。
- 应用级描述: 如果需要, 使用 setTooltip 方法设置附加描述。
- Label: 必须与一个 label 控件相关联(使用 setLabelFor 方法识别相应的文本编辑控件)。

11.17 树视图控件

树视图控件

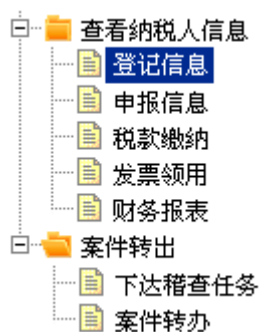


图 1:两层结构的tree

tree 视图控件用于显示层次结构的数据或文本。层次结构等级可被展开或折叠。每个 tree 节点包含一个文本和一个可展开或折叠节点的箭头图标。该节点文本也可以链接到可显示相关联数据的功能。前四层等级有不同的颜色。从第五层开始颜色与第四层保持相同。

用法

tree 包含复杂信息，使用繁琐。如果可能请避免使用而考虑选择其它控件，尤其是在 iViews 中。总之应该避免包含多于 2-3 层的 Trees。

如何避免使用 Trees

如果 tree 包含的数据（elements）很少，则该层次结构可平展为列表，各项可以遵循字母序或关联度的顺序排列。

考虑使用 下拉列表框(dropdown list boxes), shuffler (filter) 或与 table 相结合的标签栏（tabstrip）来选择部分数据集。相对那些必须滚动或必需分页显示的大型 tree，这会产生一个更简单的用户界面。

式样相关属性

树型视图控件没有 width 属性，为了设置宽度，把 tree 放入一个 grid 布局控件中。使用 title 属性 为 tree 设置标题。对于 tree items，可以定义 item 文本和相应的 tooltip 文本。(属性 text 和 tooltip).

关于 Tree 视图的更多信息

浏览器兼容性

Netscape 4.7 不能显示标准 tree 控件的某些细微差别。在 Netscape 4.7 中，tree 没有边框并且标题高度与标准 tree 不同 (图 1).

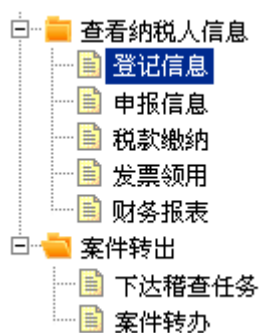


图 1: 标准 Tree

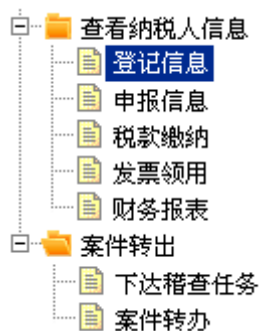


图 2: Netscape Navigator 4.7 中的 Tree

Netscape 4.7 中，tree 视图总是完全展开。在本地不可能展开和折叠节点，应用程序必须来处理这些操作(请求服务器的往返操作)

界面编辑器可编辑性

在界面编辑器中，可以修改 tree 视图控件的如下属性：

Group	Style	IE 5 and above	Netscape 4.7
Level Background Colors	Background Color of 1st Level	x	x
	Background Color of 2nd Level	x	x
	Background Color of 3rd Level	x	x
	Background Color of 4th Level	x	
Tree Icons	URL to "Expand" Icon	x	
	URL to "Collapse" Icon	x	x
	URL to "Node" Icon	x	x
Container	Height of Tree Icon	x	
	Width of Tree Icon	x	
	Background Color of Container Title	x	x
	Font Color of Container Title	x	x
	Height of Container Title	x	
	Container Border	x	
	Bottom Border of Container	x	
	Cell Padding	x	

表 1: tree 视图控件的可编辑式样

常用式样请参阅 HTMLB Controls and Style Editor in Customer Branding and Style Editor.

可访问性 - 508 支持

- 操作 (keyboard): 每个 tree 节点可以缺省插入可访问层次
- 缺省描述: 由 HTMLB 解释引擎为每个 tree 节点提供。
- 应用级描述: 如果需要的话, 使用 setTooltip 方法为每个树型节点设置附加描述。

