

ComponentOne WinForms 版本InputPanel概述

WinForms版本ComponentOne InputPanel是创建并维护数据表单的新模式。它是一个容器控件，可以放在一个Windows窗体上，以帮助您创建强大的数据输入应用程序。您可以从十九种本地提供的数据输入控件中选择进行组合，或者使用任何的三方.NET控件，仅需要简单地通过InputPanel的智能标签上的Add New Item组合框进行选择就可以完成。WinForms InputPanel提供自动的绑定，布局和样式。当您在设计时将C1InputPanel绑定到一个数据源对象时，该控件将自动产生用来查看并编辑数据源中数据的组件，包括键盘快捷键。

WinForms版本InputPanel提供一个键盘快捷键设计器，用来自动地添加或者修复重复的键盘快捷方式。使用WinForms版本InputPanel，对您的窗体进行修改将不必担心会重新构建整个窗体。全部的十九个内置的输入控件共享一组公共的样式，属性，方法以及事件的集合，这将使得使用C1InputPanel进行编程非常容易且一致。这种简单的配置搭配上低廉的维护成本将带来生产力的提高。您可以分分钟使用WinForms版本InputPanel生成优雅交互式的数据库表，请停止花费时间构建数据库表吧。

ComponentOne Studio for WinForms帮助

入门

关于安装ComponentOne Studio for WinForms, 授权, 技术支持, 命名空间和使用控件创建工程的信息，请参阅[Getting Started with Studio for WinForms](#)。

新功能

关于ComponentOne Studio for WinForms新功能，请参阅[What's New in Studio for WinForms](#)。

主要功能

WinForms版本ComponentOne 的InputPanel 提供了许多丰富的功能，列举如下：

- 简单的设置

拖拽一个C1InputPanel至您的窗体，连接到一个数据源，选择记录集合，之后每一个字段的输入组件将自动为您创建。全部的字段以及关联的标签始终保持对齐。同时将添加一个导航栏至窗体，在运行时为最终用户提供在记录之间导航的能力。

- 通用的.NET数据绑定

WinForms版本的InputPanel可以通用地绑定到任意的.NET数据源，仅需要添加很少的或者完全不必添加代码，允许您在数秒间创建一个完全的可导航的数据库浏览程序。

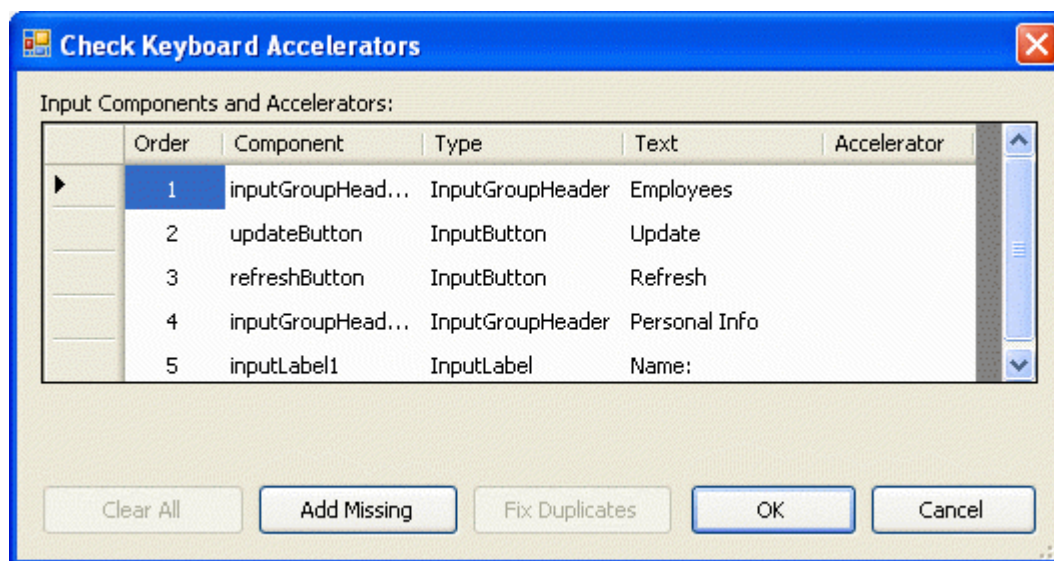
- 自动Tab键停靠顺序

对您的数据表单进行修改，无需担心需要重构整个窗体。

–添加组件并在窗体上对其进行移动，不需要丢弃整个窗体的结构。WinForms 版本 InputPanel在设计时，WinForms 版本 InputPanel所以，在运行时，用户可以使用Tab键从一个条目到另一个条目，这是非常快速并且友好的编辑体验。

- 快捷键

当窗体通过来自于数据源的字段产生时，其快捷键是自动生成的。InputPanel提供了一个快捷键编辑器以管理任何新增加的快捷键，同时可以消除任何重复。



- 通过一个属性调整布局

您可以毫不费力地调整布局。默认情况下，数据将自动地按照一行一行的方式进行流布局。您可以通过设置一个属性打破流布局方式。例如，您可以将单行的流布局方式打破为一个两列布局的方式。仅需要通过修改一个属性，新的布局将显示两列，并且组件将保持对齐。如果没有WinForms 版本InputPanel的流布局管理

您必须一次移动一个组件，并且在最后调整全部组件的对齐方式。

- 使用Microsoft Office 2007 样式增强您的界面

WinForms版本InputPanel支持模仿Office 2007可用的视觉样式，包括Blue，Silver以及Black。

- 以图形效果和组件交互

运行时每一个字段的外观是可定制的，包括移动鼠标悬停在组件上方，按下，或者禁用控件的不同状态。对于这个图形效果，不需要任何额外的步骤。

- 滚动条自动按需添加

当窗体改变尺寸大小时，WinForms版本InputPanel将按照需要自动地添加水平和垂直滚动条。此功能在设计时和运行时均可用。

- 验证和错误处理

当用户输入无效的输入时，将自动提供一个可定制的可视化警告。一个红色的框将出现在输入组件周围。您还可以添加自定义的警告，如显示错误提示。例如，您可以允许最终用户仅能够输入字母字符，当用户输入其它类型的字符时，显示一个错误指示。

- 在同一个编辑器中管理您的任务

InputPanel允许您在面板中对各个项目进行重新排布，添加新的项目，以及对项目进行格式化。为了有效地管理这些任务，**InputPanel**提供了一个集合编辑器。该集合编辑器可以很容易从上下文菜单或智能标签进行访问。

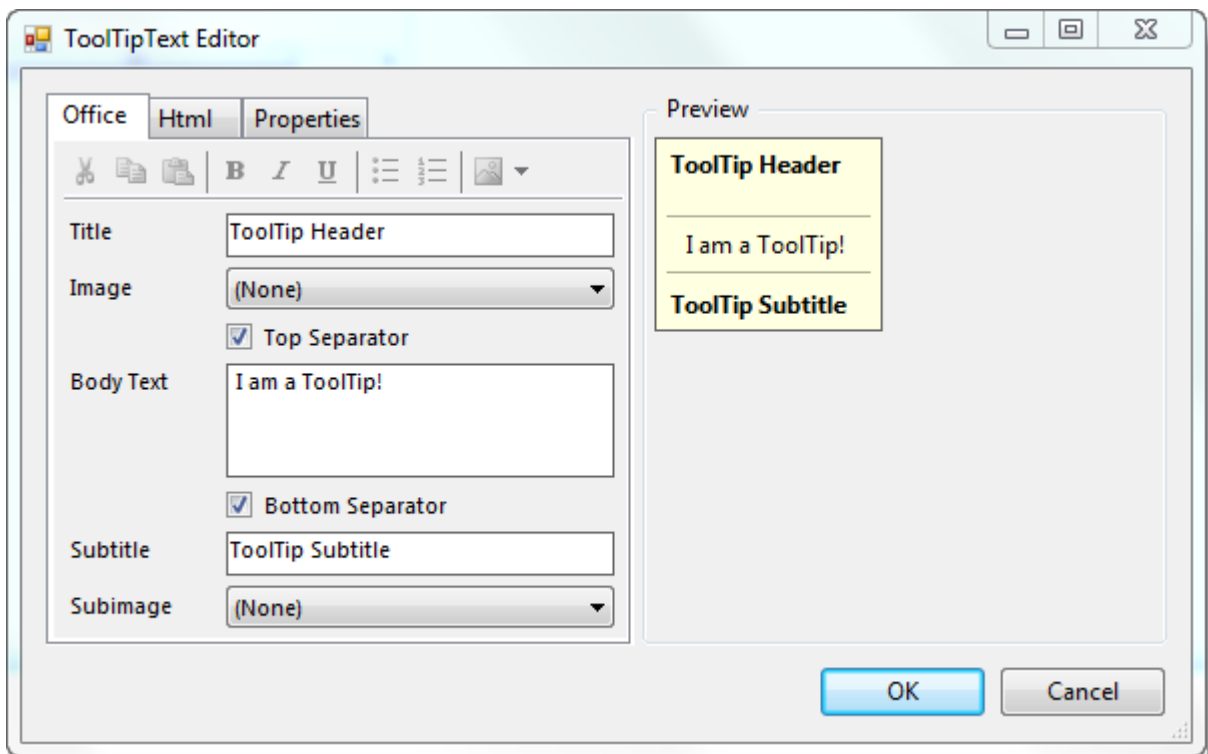
设计器允许您为面板选择数据源，创建全部的输入组建，并可以编辑单独的组件（添加/移除/设置属性）。关于设计器的更多信息，请参见**C1InputPanel** 设计器。

- 在窗体上排布组件所占的空间

安排你的窗体上的组件的空间，只需调整组件的**Padding**。您可以自由地增加或减少组件之间的间距，此时组件将保持整齐的对齐。您也可以通过使用**HorizontalAlign**或者**VerticalAlign**达到同样的目的。

- 添加工具提示以提升用户体验

在设计时，您可以非常容易地添加图形元素，比如说工具提示。**ToolTipText**属性有一个编辑器，在此您可以输入您的工具提示文本，或者添加图像，甚至支持**HTML**格式。



- 设计时的简单数据绑定

C1InputPanel 控件具有一个**DataSource**属性，可以在设计时进行设置，并能够自动的生成数据条目内置的组件。

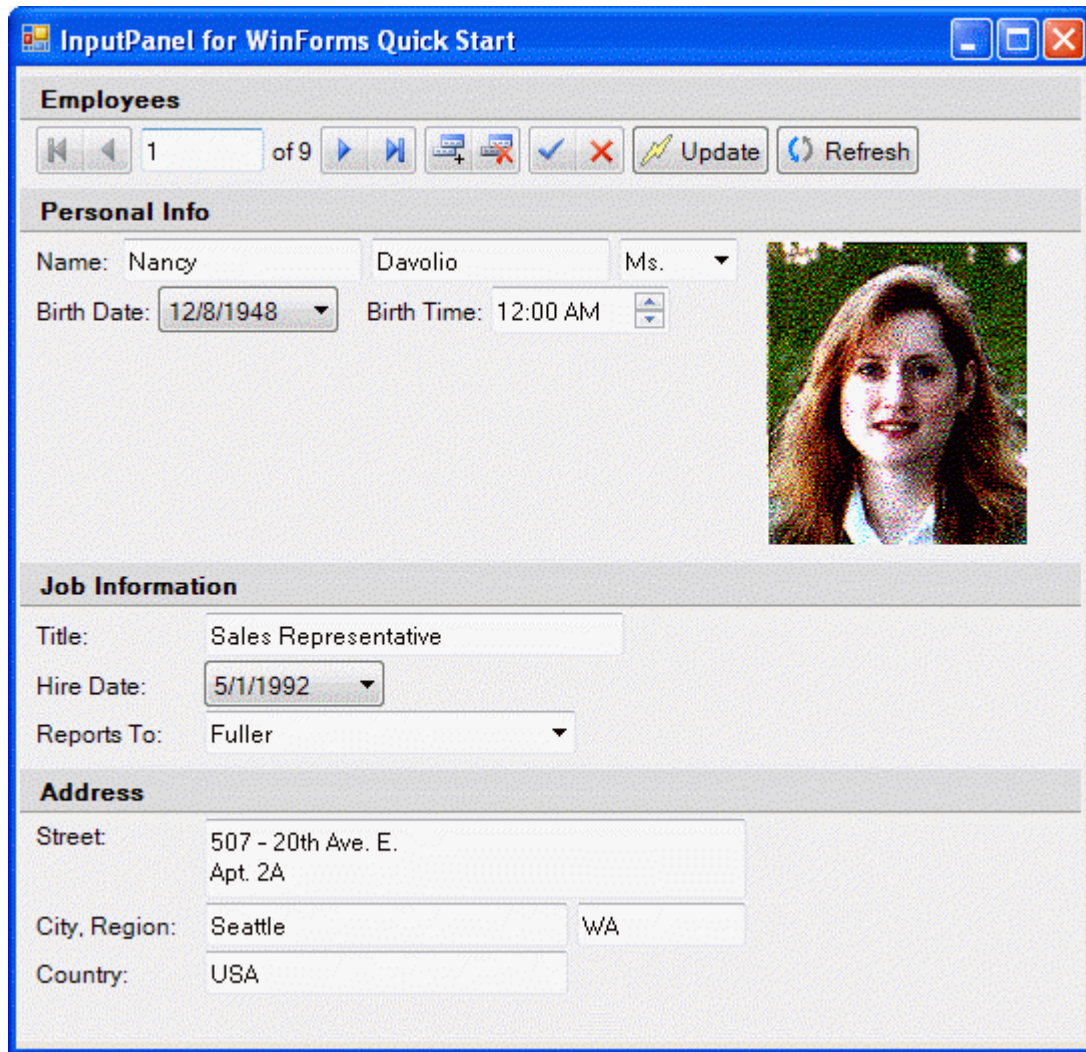
- 在面板内简单，快速，灵活的对单个输入组件进行流布局

C1InputPanel 提供在面板内简单，快速，灵活的对单个输入组件进行流布局。其基本原理是一个具有可选的分隔符的流布局。在布局中，面板将控制焦点/**tab**键顺序，等等。

WinForms版本InputPanel快速入门

本章节提供了一个构建一个提供输入每一个员工的姓名、头衔、称谓、生日、雇佣日期、地址、城市、区域、邮编、地区、家庭电话、分机号、照片、备注以及字段报表的一个详细步骤指导。应用程序使用分组页眉将员工信息分成三类：员工的出生日期和姓名，工作职责和职位，以及员工的地址。

完成这个快速入门将生成一个能够输入的应用程序，如下图所示：



第一步：为C1InputPanel创建一个数据源

在此步骤中，您将创建一个数据源，并在之后通过C1InputPanel 集合编辑器将其绑定到InputPanel的每一项。创建一个.NET工程，并完成以下步骤：

添加新的数据源

1. 在工程的工具栏上，从数据菜单选择添加新的数据源。将出现数据源配置向导对话框。
2. 选择数据库并单击下一步。
3. 单击“新建连接”。
4. 在“添加连接”对话框中，单击“浏览”。
5. 在选择微软Access数据库文件对话框，查找到Nwind.mdb (Vista / 7: C:\Users\\Documents\ComponentOne Samples\Common, In XP:C:\Documents and Settings\\My Documents\ComponentOne Samples\Common)，单击打开，接下来单击OK。
6. 单击“下一步”按钮继续。将出现一个对话框，询问您是否要添加数据文件到您的项目并修改连接字符串。因为

不必将数据库复制到您的工程，请单击No。

7. 确保“**Yes, save the connection as**”复选框为选中状态，并单击下一步以继续。
8. 展开**Tables**节点，并选择**Employees**对象。
9. 单击“完成”。一个数据集和连接字符串将添加到您的项目。
10. 在**Visual Studio** 工具栏上展开数据分组，并双击**BindingSource** 将其添加到您的组件托盘中。
11. 从属性窗体选择**bindingSource1**，设置其**DataSource**属性为**nwindDataSet**，设置其**DataMember**属性为**Employees**。

一个**employeesTableAdapter** 将添加至组件托盘，同时以下代码将添加到**Form_Load**事件的处理函数：

Visual Basic

Visual Basic

```
Me.EmployeesTableAdapter.Fill(Me.NwindDataSet.Employees)
```

C#

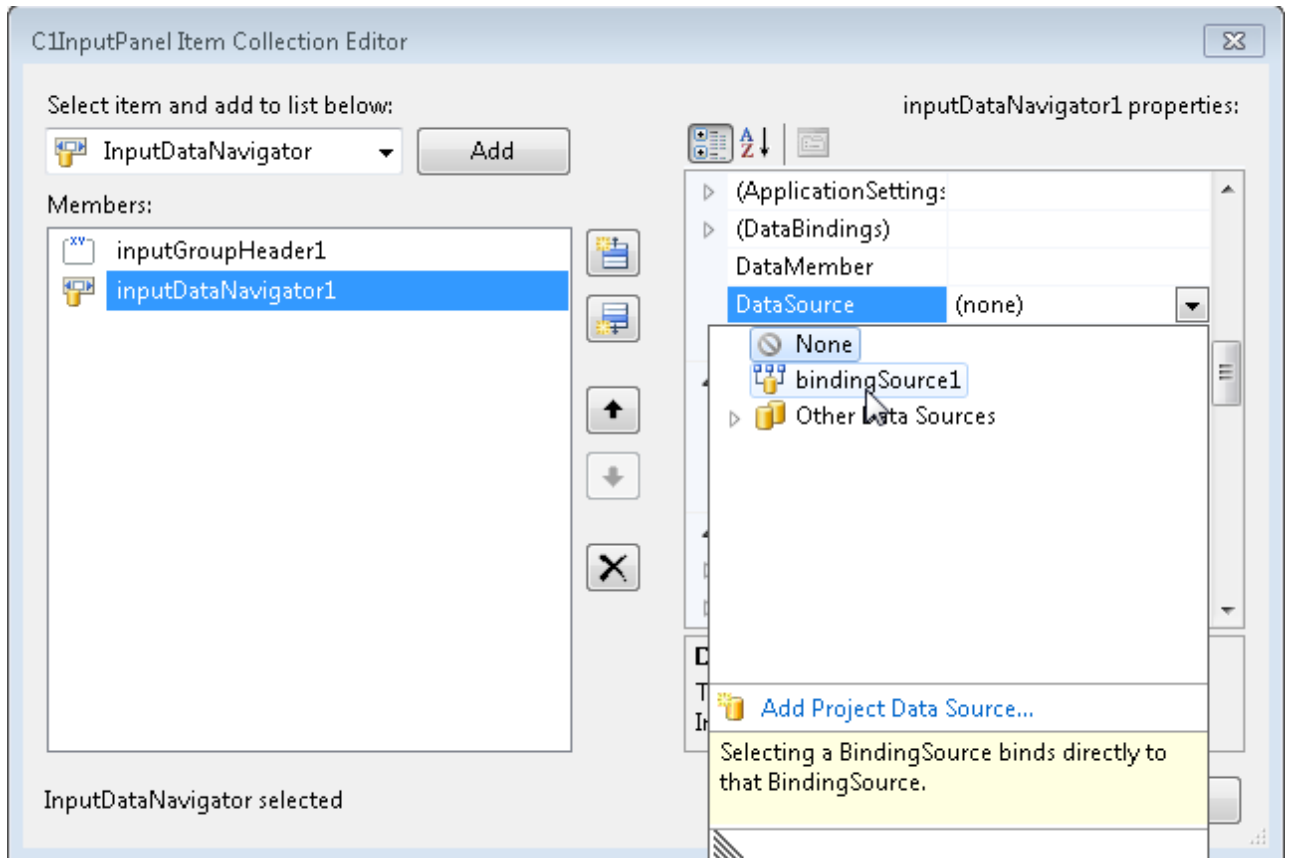
C#

```
this.EmployeesTableAdapter.Fill(this.NwindDataSet.Employees);
```

第二步：通过编辑器添加第一组输入组件

为通过**C1InputPanel**项目集合编辑器添加并格式化输入组件，请完成以下任务：

1. 调整**Form1**的尺寸为**534x466**。
2. 向窗体添加**C1InputPanel**控件，并设置其**Size**属性的值为**534x466**，同时设置**SmoothScaling** 属性的值为**True**。
3. 右键单击**C1InputPanel** 控制并从菜单中选择编辑项目。将会出现**C1InputPanel**项目集合编辑器。
4. 单击添加以添加**InputGroupHeader**组件至**C1InputPanel**，并设置其**Text**属性的值为“**Employees**”。
5. 从**C1InputPanel** 项目集合编辑器的下拉列表框中选择**InputDataNavigator**并单击添加。
6. 设置其**DataSource**属性为“**bindingsource1**”以及其**InputComponent.Break** 属性为**None**，因此下一个添加的元素会紧随其后。

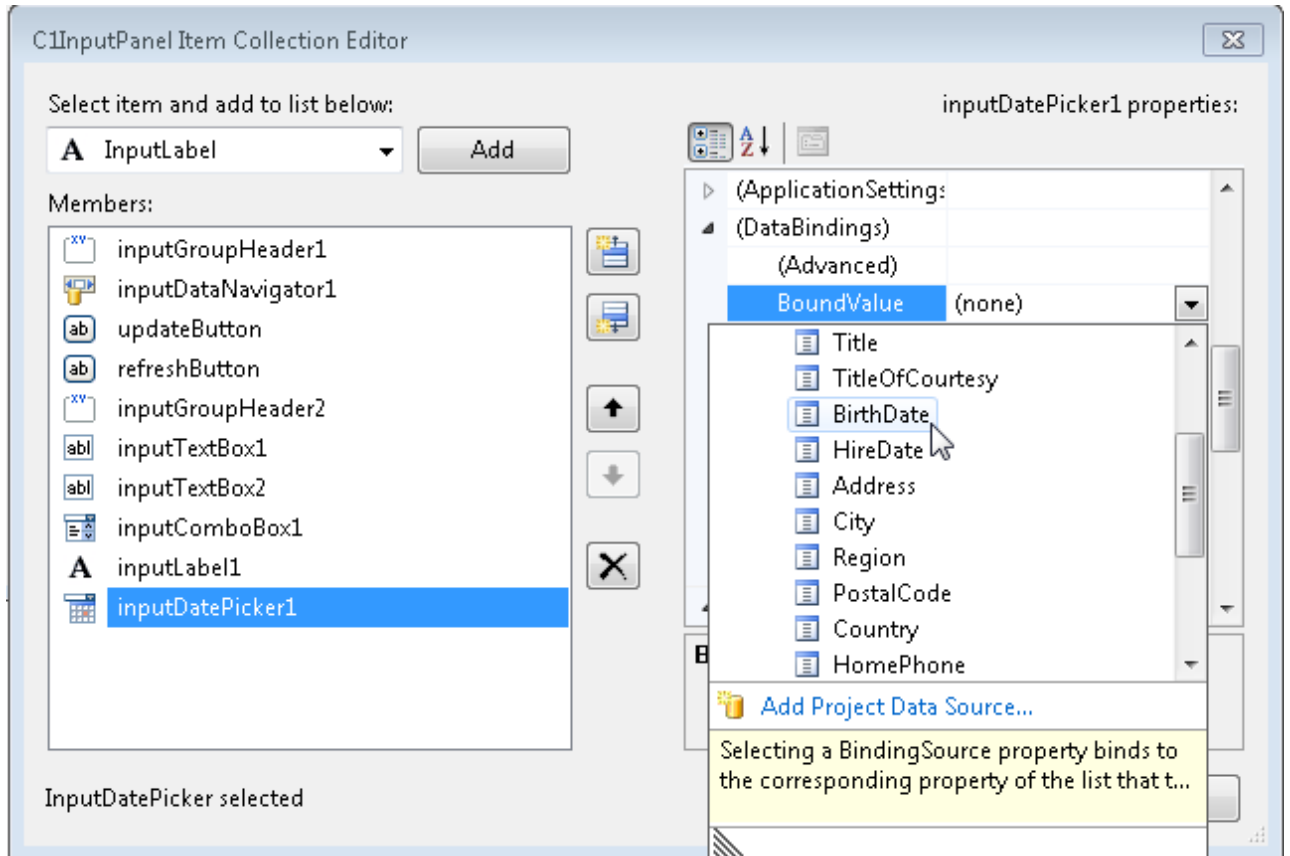


7. 从下拉列表框中选择InputButton，并单击添加。
8. 设置此InputButton的Name为updateButton，Break属性的值为None，因此下一个输入元素将紧随其后，之后设置其Text属性的值为“Update”。
9. 单击InputButton的Image属性以浏览希望添加到按钮的图像。
10. 从下拉列表框中选择InputButton，并单击添加。
11. 设置InputButton1的Name为refreshButton，并设置其Text为“Refresh”。

第三步：通过编辑器添加第二组的输入组件

为了使用C1InputPanel 项目集合编辑器向InputPanel添加第二组的输入组件，需要完成以下步骤：

1. 从下拉列表框选择InputGroupHeader，单击Add，并设置Text 属性的值为“Personal Info”。
2. 从下拉列表框选择InputLabel，并设置Text属性的值为“Name:”。
3. 从下拉列表框选择InputTextBox，并单击Add。
4. 设置InputTextBox1的Break属性为None，因此下一个输入元素将紧随其后。展开其DataBindings节点，并设置其文本为“bindingSource1-FirstName”。
5. 从下拉列表框中选择InputTextBox，并单击Add。展开其DataBindings 节点，并设置其Text 为“bindingSource1-LastName”。
6. 从下拉列表框中选择InputComboBox，并单击Add。展开其DataBindings节点，并设置其Text 为“bindingSource1-TitleOfCourtesy”。
7. 从下拉列表框中选择InputLabel，并单击Add。设置其Text属性为“Birth Date:”。
8. 从下拉列表框中选择InputDatePicker，并单击Add。设置其Break属性为None，并展开其DataBindings节点，之后设置其BoundValue属性的值为“bindingSource1-BirthDate”。



9. 从下拉列表框选择InputLabel，并单击Add。设置其Text属性的值为“Birth Time:”。
10. 从下拉列表框选择InputTimePicker并单击Add。展开其DataBindings节点并设置其BoundValue属性为“bindingSource1-BirthDate”，同时设置其Break属性的值为Column。
11. 从下拉列表框选择InputImage，并单击Add。展开其DataBindings节点并设置其Image属性的值为“bindingSource1-Photo”。

第四步：通过编辑器添加第三组输入组件

为了通过C1InputPanel项目集合编辑器向InputPanel添加第三组的输入组件，需完成以下步骤：

1. 从下拉列表框中选择InputGroupHeader并单击Add。设置其Text属性的值为“Job Information”。
2. 从下拉列表框中选择InputLabel并单击Add。设置其Text属性的值为“Title:”。
3. 从下拉列表框中选择InputTextBox，并单击Add。展开其DataBindings节点并设置其Text属性的值为“bindingSource1-Title”。
4. 从下拉列表框中选择InputLabel并单击Add。设置其Text属性的值为“HireDate:”。
5. 从下拉列表框中选择InputDatePicker，并单击Add。展开其DataBindings节点，并设置其Value属性的值为“bindingSource1-HireDate”。设置其ValueIsNull属性的值为False。
6. 从下拉列表框中选择InputLabel并单击Add。设置其Text属性的值为“Reports To:”。
7. 从下拉列表框中选择InputComboBox，并单击Add。展开其DataBindings节点并设置其SelectedValue属性的值为“bindingSource1-ReportsTo”，DataSource属性的值为“BindingSource1”，DisplayMember属性的值为“LastName”，FormattingEnabled属性的值为“True”，ValueMember属性的值为“EmployeeID”。

第五步：通过编辑器添加第四组的输入组件

为了通过C1InputPanel项目集合编辑器向C1InputPanel添加第四组的输入组件，需完成以下步骤：

1. 从下拉列表框中选择InputGroupHeader并单击Add。设置其Text属性的值为“Address”。
2. 从下拉列表框中选择InputLabel并单击Add。设置其Text属性的值为“Street”。

3. 从下拉列表框中选择InputTextBox，并单击Add。设置其Multiline属性的值为True。展开其DataBindings节点，并设置其Text属性的值为“bindingSource1- Address”。
4. 从下拉列表框中选择InputLabel并单击Add。设置其Text属性的值为“City,Region”。
5. 从下拉列表框中选择InputTextBox 并单击Add。展开其DataBindings节点并将其Text属性的值设置为“bindingSource1-City”。设置Break属性的值为None。
6. 从下拉列表框中选择InputTextBox 并单击Add。展开其DataBindings 节点并设置其Text属性的值为“bindingSource1-Region”。
7. 从下拉列表框中选择InputLabel并单击Add。设置其Text属性的值为“Country:”。
8. 从下拉列表框中选择InputTextBox 并单击Add。展开其DataBindings 节点并设置其Text属性的值为“bindingSource1-Country”。

您已成功绑定C1InputPanel全部的元素至数据源。

按F5键运行您的应用程序。

为观察变化，需完成以下任务：

- 单击位于InputDataNavigator上的Move Next按钮。全部的输入字段将发生更新。

恭喜！

您已成功创建了一个数据源，用来绑定输入组件。此外，您还学习到了如何使用Break属性控制输入组件的布局。


设计时支持

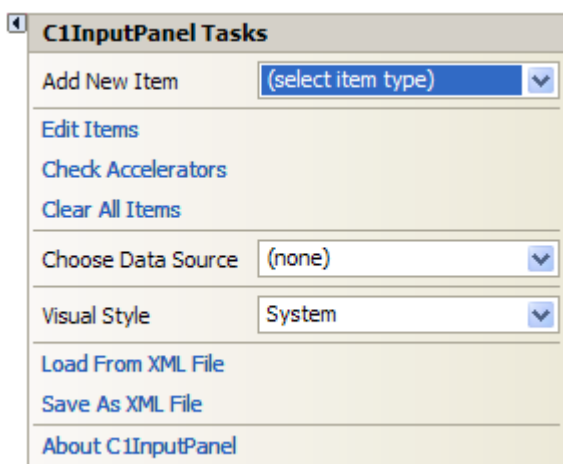
C1InputPanel提供了定制的上下文菜单，智能标签，以及设计器，这些提供了丰富的设计时支持并简化了对象模型的使用。

以下主题介绍如何使用C1InputPanel的设计时环境配置C1InputPanel组件。

C1InputPanel 智能标签

在Visual Studio中，每一个位于WinForms版本ComponentOne InputPanel 的控件都包含一个智能标签。智能标签是一个任务菜单的快捷方式，提供每一个控件最常用的属性。

为访问C1InputPanel任务菜单，单击C1InputPanel控件右上角的智能标签（）。这将打开C1InputPanel任务菜单。



C1InputPanel任务菜单操作如下：

添加新项

选择Add New Item下拉框提供一个可供选择的输入组件的列表。

编辑项目

点击编辑项目打开C1InputPanel项目集合编辑器。

检查加速键设置

单击检查加速键设置打开检查加速键设置设计器。

清除所有项目

点击清除所有项目从C1InputPanel删除项目。

选择数据源

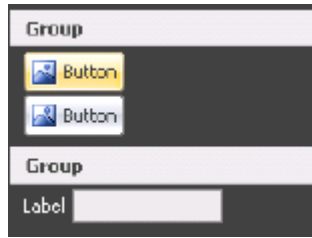
选择选择数据源下拉框，打开可用的数据源列表，同时也允许您添加新的数据源。为了向工程添加一个新的数据源，单击添加工程数据源以打开数据源配置向导。

视觉样式

选择视觉样式下拉框，打开一个包含几个内置的InputPanel方案的列表框供您选择。默认的视觉样式是System。C1InputPanel提供了一个Custom视觉样式，在此您可以创建您自己的VisualStyle或者您可以从以下视觉样式进行选择：

Visual Style	Image
Name	

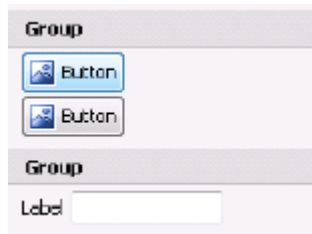
Office2007Black



Office2007Blue



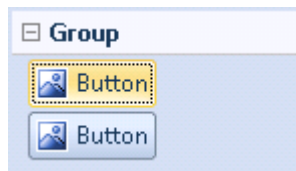
Office2007Silver



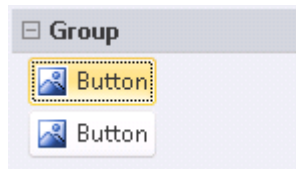
Office2010Black



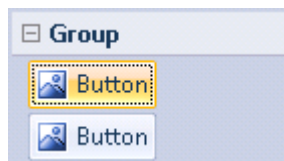
Office2010Blue



Office2010Silver



Windows7



System (the

default visual style
for the
C1InputPanel
control)



从XML文件加载

选择从XML文件加载选项将打开从XML文件加载对话框，您可以用来浏览并为InputPanel的布局选择一个XML文件。

另存为XML文件

选择另存为XML文件选项将打开另存为XML文件对话框，您可以用来浏览并为InputPanel的布局选择一个XML文件。

关于C1InputPanel

单击About项目显示关于C1InputPanel对话框，它可以帮助发现C1InputPanel的版本号以及在线资源。

C1InputPanel 上下文菜单

C1InputPanel具有Designer Verb，在快捷键菜单或者C1InputPanel组件关联的上下文菜单中显示。

C1InputPanel 上下文菜单

C1InputPanel具有Designer Verb，在快捷键菜单或者C1InputPanel组件关联的上下文菜单中显示。
C1InputPanel上下文菜单

在C1InputPanel控件上右键单击任何位置可显示其上下文菜单。

下表提供了一个对C1InputPanel所添加的自定义命令的简短描述：

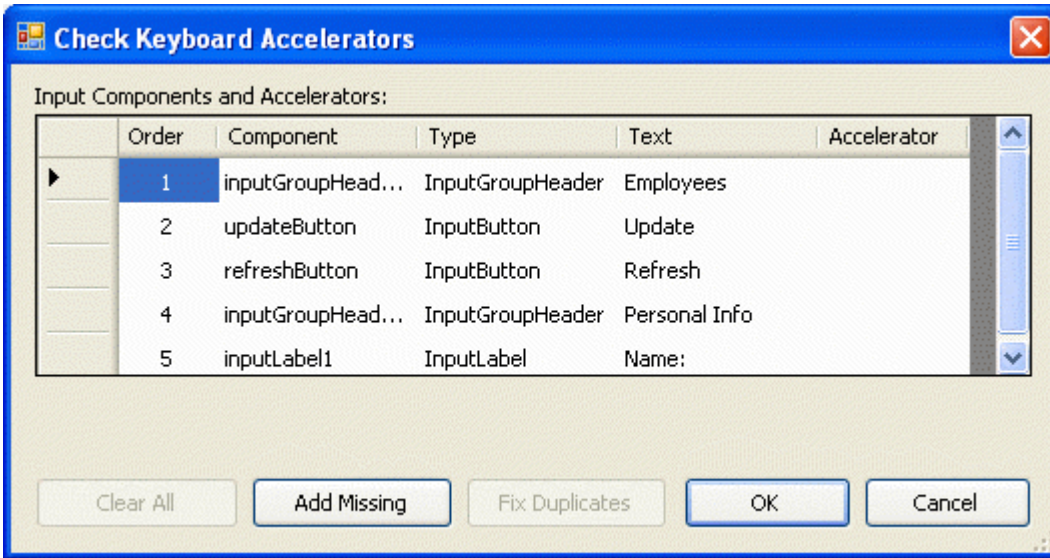
命令	描述
关于 ComponentOne C1InputPanel...	显示关于ComponentOne C1InputPanel对话框，这可以帮助您查找C1InputPanel的版本号以及其他的在线资源。
编辑项目	点击编辑项目打开C1InputPanel项目集合编辑器。
检查键盘加速键	单击检查键盘加速键设置打开检查加速键设置设计器。
清除所有项目	点击清除所有项目从C1InputPanel删除项目。

检查键盘加速键

当您添加了一个输入组件至C1InputPanel，它会自动的添加到检查键盘加速键设计器。

访问检查键盘加速键

- 右键单击C1InputPanel 并选择检查键盘加速键。



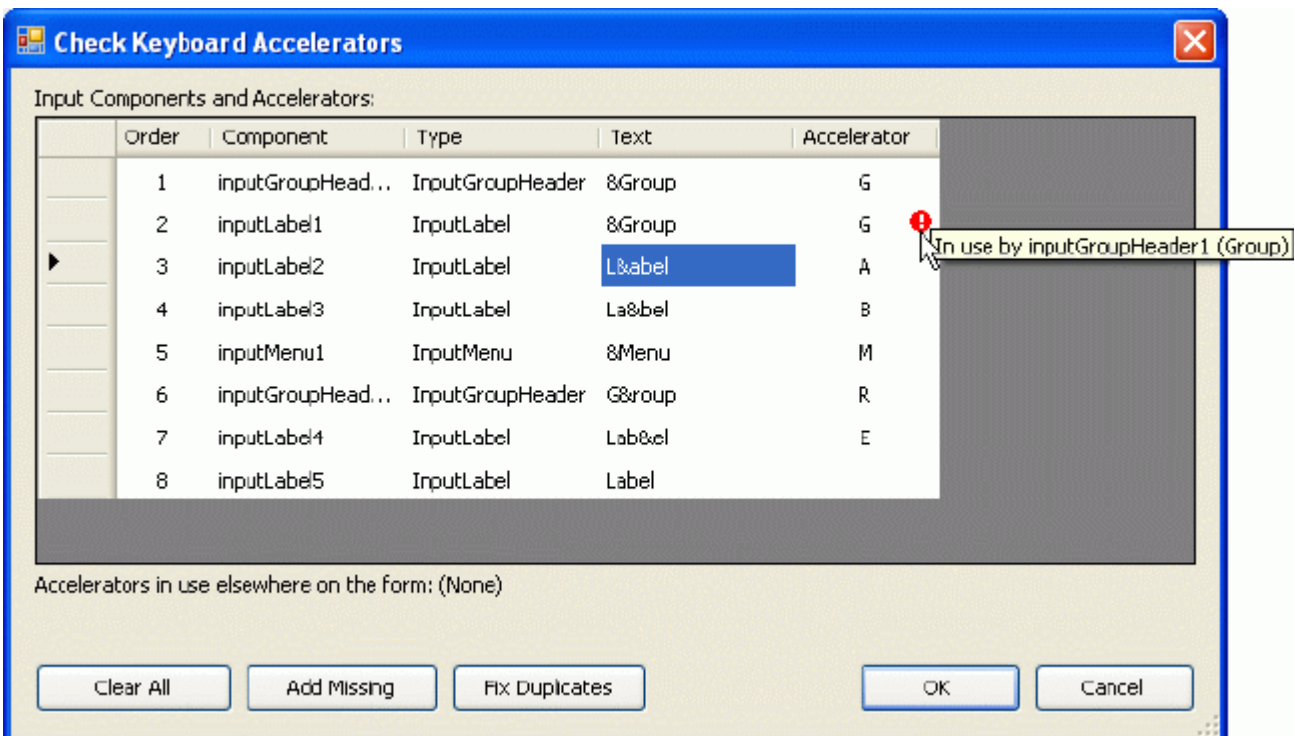
为全部的输入组件自动添加键盘加速键

- 单击位于检查键盘加速键设计器上的Add Missing按钮

通常情况下，每一个输入控件的Text属性的第一个字符将作为默认的键盘加速键。

注意：如果您有一些Text属性属性的值相同的几个输入组件，那么自动添加缺失的加速键方法将从Text名称中为每一个输入组件选取一个不同的字符。如果Text名称中的每一个字符都已经被分配，则将不会指定任何的键盘加速键。您将需要为其在键盘加速键设计器中输入一个唯一的键值做为加速键。

如果您键入了一个重复的键盘加速键，则将会显示一个带有警告信息的错误图像。



单击Fix Duplicates按钮将自动地修复全部重复的键盘加速键值，通过为每一个重复的键值指定一个全局唯一的键盘加速键。

C1InputPanel 设计器

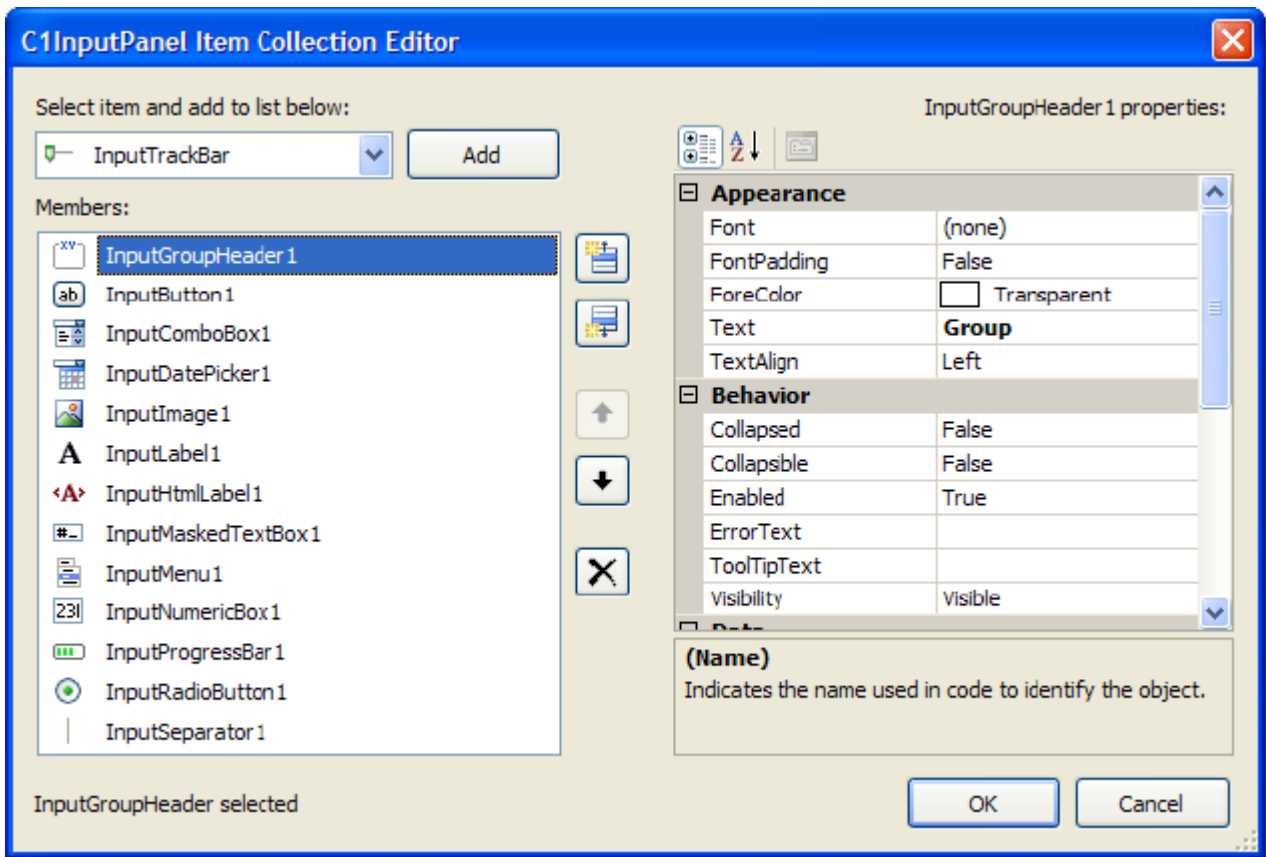
C1InputPanel，具有一个强大但是简单易用的设计器，C1InputPanel Item Collection Editor，用来添加，删除，或者编辑面板上的输入组件项目。该编辑器应用程序的主要部分由一个Windows窗体构成，方便用户修改输入组件的属性。编辑器的命令按钮允许用户快速地通过一次单击操作在选中的项目之前添加或插入项目，移除或者移动输入组件的位置。每次您通过设计器添加一个输入组件，该输入组件将自动地出现在窗体上的C1InputPanel中。当您添加一个输入组件，其属性将出现在编辑器的分组面板，因此您可以方便地修改输入组件的属性。

C1InputPanel的WYSIWYG（所见即所得）设计器界面使得您在设计时查看各个元素显示在InputPanel上的结果变得非常容易，完全不需要运行工程查看运行时的结果。

访问C1InputPanel 项目集合编辑器



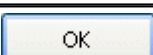
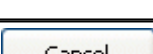
- 右键单击C1InputPanel 并选择编辑项目。

将出现的C1InputPanel 项目集合编辑器。



下表提供了C1InputPanel项目集合编辑器全部命令按钮功能的描述：


命令按钮	描述
	将组合框选中的项目添加到C1InputPanel最后一个输入项目之后。
	将组合框中选中的项目添加到Members列表框选中的项目之前。
	将组合框中选中的项目添加到Members列表框选中的项目之后。
	将Members列表框中选中的输入项目向上移动一个位置。


	将Members列表框中选中的输入项目向下移动一个位置。
	从C1InputPanel的Members列表框中删除选中的输入项目。
	保存并更新C1InputPanel所做的修改并关闭C1InputPanel项目集合编辑器。
	取消您所做的更改并关闭C1InputPanel 项目集合编辑器。

添加输入组件

为添加一个输入组件，从下拉列表框选择一个项目并单击Add按钮。

在当前输入组件之前或之后添加一个输入组件

单击Insert Before  ,按钮以便在当前/选中的输入组件之前插入一个输入组件。

单击Insert After  ,按钮以便在当前/选中的输入组件之后插入一个输入组件。

改变输入组件的位置

为改变一个输入组件在C1Input组件中的位置，单击MoveUp  以及MoveDown  , MoveUp按钮将选中的项目在CInputPanel中向上移动一个位置，而MoveDown按钮将选中的项目在C1InputPanel中向下移动一个位置。

删除一个特定的输入组件

从Members列表框中选择您希望删除的输入项目并单击Delete按钮。

编辑输入组件

一旦一个输入组件添加到编辑器中的Members列表框，您可以在Property Grid中间编辑其属性。为了编辑一个特定的输入组件，从Members列表框中选中该输入组件并在Property Grid中间编辑其属性。

ErrorText 编辑器

ErrorText编辑器和ToolTip编辑器相似，仅用做创建基于HTML的消息框，该消息框将在用户输入了C1InputPanel项目无法接受的数据时显示错误信息。

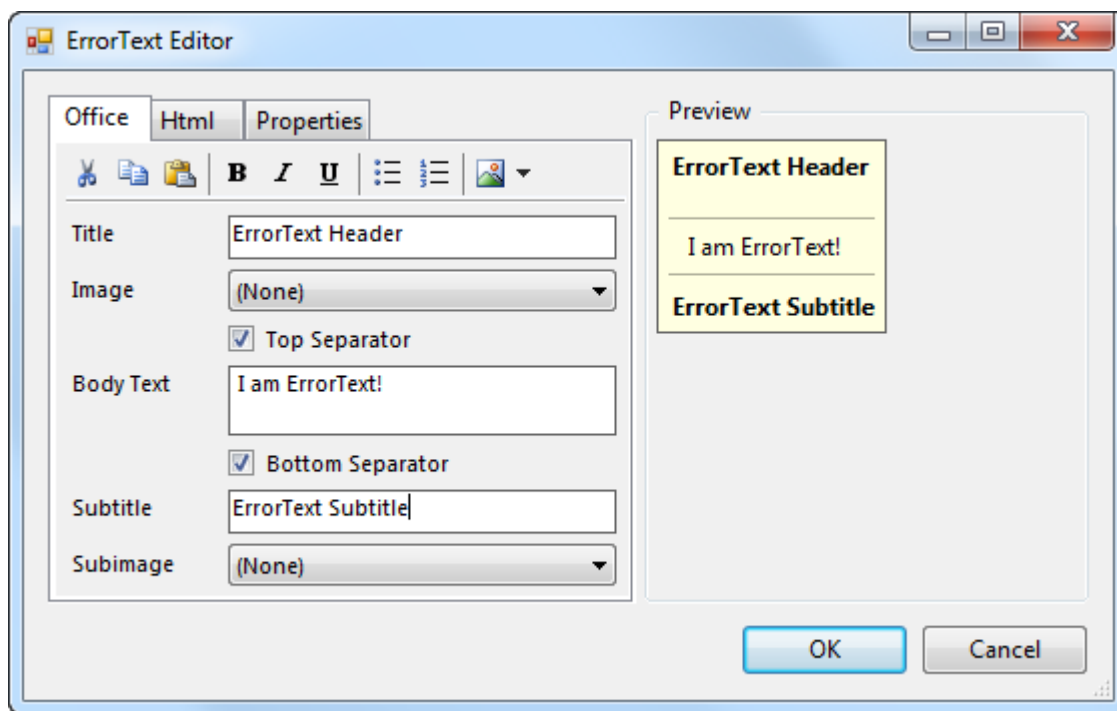
访问ErrorText编辑器

为打开ErrorText编辑器，请遵循以下步骤：

1. 单击C1InputPanel智能标签打开C1Input任务菜单，然后选择编辑项目。
2. 在Memebers面板中，选择一个组件。该组件的属性列表将出现在Property Grid中。
3. 在PropertyGrid中，查找ErrorText属性并按下其省略号 按钮。

ErrorText编辑器布局

当打开ErrorText编辑器时，您可以看见三个面板，Office，HTML，以及Properties，位于对话框右侧。每一个标签页包含一个编辑器，在此您可以修改错误信息的内容，源，或者属性。在编辑器的右侧是预览面板，这里提供了一个最终所显示的错误消息的实时预览。

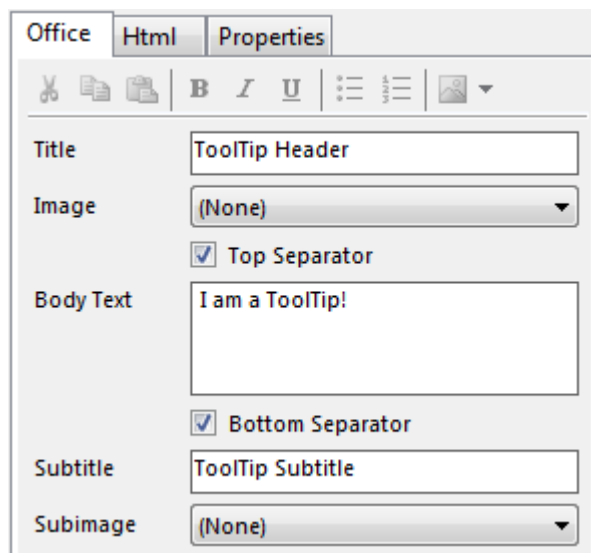


以下主题将详细讲述Office，HTML，以及Properties标签页的功能。

Office标签页

Office标签页用作添加内容，图形，以及分隔线至您的错误消息。编辑器自动地创建错误消息底层全部的HTML代码，以节约您的时间和精力。在Office标签页，您对错误消息所做的任何添加或者修改操作将立刻反应在预览面板。

Office标签页同样具有一个包含命令按钮的工具栏。您可以通过命令按钮以添加元素至您的工程。



下表描述了Office标签页中的元素：

元素	描述
标题	标题元素出现在错误消息的顶部，默认情况下，将显示为粗体。您可以修改该标题的HTML内容。
图像	该图像将出现在错误消息的正文之上。默认情况下，将显示在左侧。
顶部分隔	在标题和/或图像之下放置一个水平标尺。该水平标尺为深灰色。

线

正文文本 正文文本包括错误消息的主体内容部分；它包含为工具出现的指令或者描述。它将出现在标题下方副标题的上方。

底部分隔线 放置一个水平标尺在正文文本下方。该水平标尺颜色为深灰色。

副标题 该副标题元素出现在错误消息的底部。

Subimage 该图像将出现在错误消息正文下方。默认情况下，将显示在左侧。

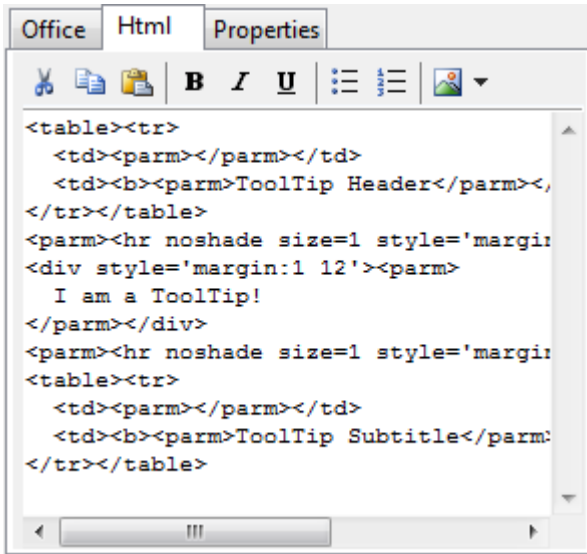
下表提供了Office标签页全部命令按钮的描述：

按钮	名称	描述
	剪切	从HTML编辑器窗体移除选中的内容，并将其放置在剪贴板中。
	复制	从HTML编辑器窗体复制选中的内容至剪贴板。
	粘贴	将剪贴板中的内容粘贴至HTML编辑器窗体。
B	粗体	向选中的文本添加粗体文本（）HTML标记。
I	斜体	向选中的文本添加斜体文本（<i></i>）HTML标记。
U	下划线	向选中的文本添加带下划线文本（<u></u>）HTML标记。
	项目符号列表	添加项目符号列表标签（）至HTML编辑器。
	编号列表	添加编号列表标签（）至HTML编辑器。
	插入图片	通过HTML标签插入一个图像至工具提示（）。







HTML标签

HTML标签页用作添加或这修改您的错误消息的内容。当您在HTML标签页创建一个错误消息时，您可以获得对错误消息设计的更多控制。您在HTML标签页上对错误消息所做的任何更改都将反映在预览面板。

HTML标签页由一个包含命令按钮的工具栏以及一个HTML编辑器窗体组成。您可以通过命令按钮添加标记标签至您的工程，或者您可以在HTML编辑器窗体手工敲入代码。

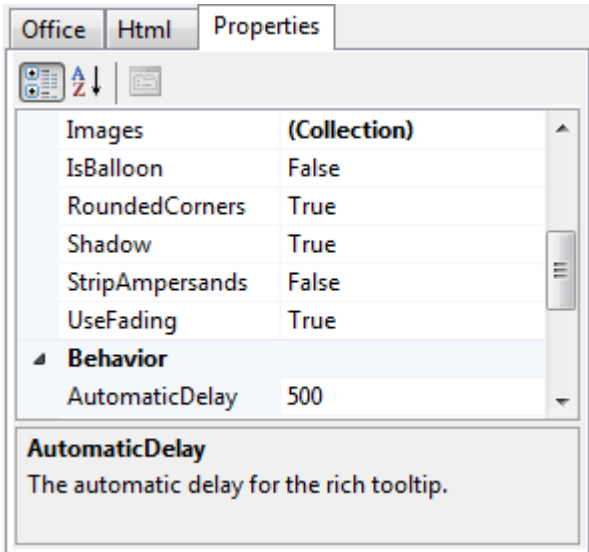


下表提供了HTML标签页所有的命令按钮的描述:

按钮 名称	描述
 剪切	从HTML编辑器窗体移除选中的内容，并将其放置在剪贴板中。
 复制	从HTML编辑器窗体复制选中的内容至剪贴板。
 粘贴	粘贴来自于剪贴板内容至HTML编辑器窗口。
B 粗体	向选中的文本添加粗体文本 () HTML标记。
<i>I</i> 斜体	向选中的文本添加斜体文本 (<i></i>) HTML标记。
<u>U</u> 下划线	向选中的文本添加带下划线文本 (<u></u>) HTML标记。
 项目符号列表	添加项目符号列表标签 () 至HTML编辑器。
 编号列表	添加编号列表标签 () 至HTML编辑器。
 插入图片	通过HTML标签插入一个图像至工具提示 ()。

“属性”标签

PropertyGrid位于属性标签页，用做自定义您的错误消息的外观和行为。您对错误消息所做的许多更改都将显示在预览面板中。



下表提供了一个影响错误消息外观的属性列表：

属性	描述
BackColor	设置错误消息窗体的背景色。
BackgroundImage	设置显示在错误消息窗体的背景图片。
BackgroundImageLayout	设置背景图像布局为None（默认），Tile，Center，Stretch，Zoom或者TileStretch。
Border	设置是否错误消息窗体应当显示一个实线边框。
BorderColor	设置错误消息窗体边框的颜色。
Font	设置错误消息内部使用的默认字体。
ForeColor	设置错误消息窗体的前景色。
Images	获取可能显示在错误消息中的图像的集合。
IsBalloon	设置是否错误消息显示为气球形状。如果设置为True，错误消息将变化为气球形状的对话框。
RoundedCorners	设置是否错误消息窗体具有圆角边框。
Shadow	设置是否错误消息窗体显示下拉阴影。
StripAmpersands	设置是否文本中间的&符号应当显示。
UseFading	设置是否在现实一个工具提示时，使用淡出效果。

下表提供了影响错误消息行为的属性列表：

属性	描述
Active	设置一个值，该值指示是否当前错误消息为活动状态。
AutomaticDelay	设置错误消息自动显示的延迟时间。
AutoPopDelay	设置错误消息在鼠标指针悬停时持续显示的时间（单位是毫秒）。
HitTestVisible	指定是否错误消息可以响应鼠标消息。
InitialDelay	设置错误消息显示之前的等待时间。
MaximumWidth	设置错误消息的最大宽度。
Opacity	获取或设置错误消息的不透明度。
ReshowDelay	设置当鼠标指针从一个组件移动到另一个组件时，在下一个错误消息出现之前必须经过的等待时间。
ShowAlways	设置一个值，该值指示是否当其父组件为非活动状态时，显示一个错误消息窗

体。

ToolTip编辑器

ToolTip编辑器用来为您的C1InputPanel创建富文本格式的工具提示。

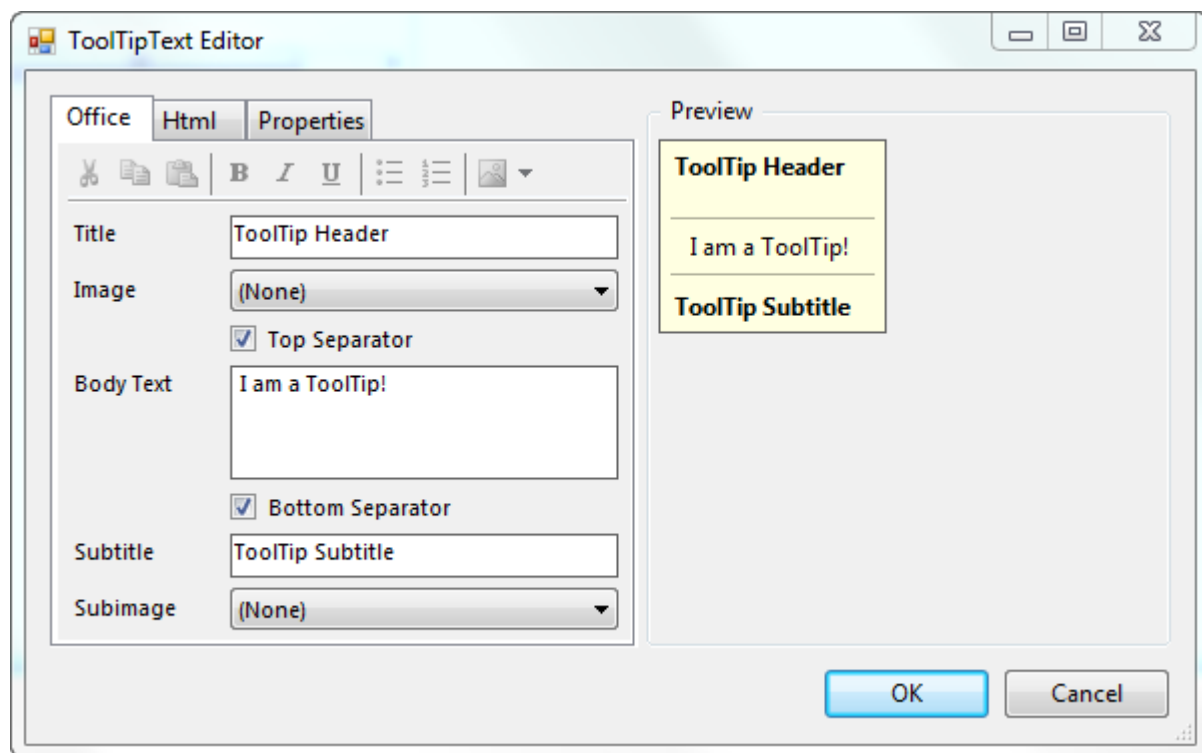
访问ToolTip编辑器

为打开ToolTip编辑器，请遵照以下步骤：

1. 单击C1InputPanel智能标签打开C1Input任务菜单，然后选择编辑项目。
2. 在Members面板中，选择一个组件。该组件的属性列表将出现在PropertyGrid中。
3. 在PropertyGrid中，查找ToolTipText属性并单击其省略号按钮。将出现ToolTip编辑器。

ToolTip编辑器的布局

当ToolTip编辑器打开时，您将看到三个不同的标签页-Office，HTML，以及属性，这些标签页位于对话框的左侧。每一个标签页包含一种编辑器的功能，您可以在这里改变内容，source或者一个ToolTip的属性。在编辑器的右侧是预览面板，它提供了一个对ToolTip最终展现的实时预览。

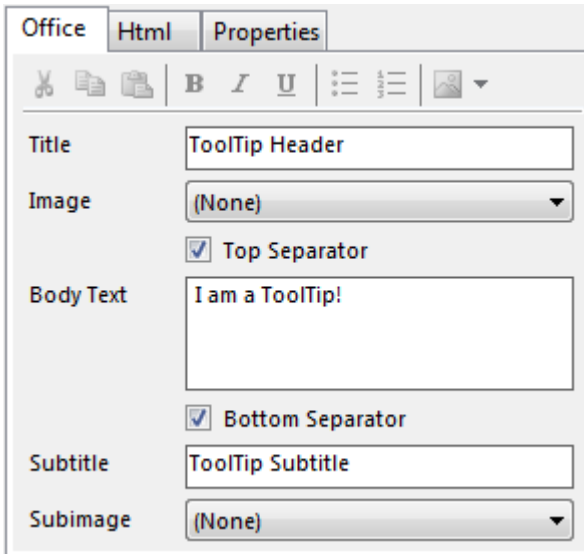


以下主题将详细讲述Office，HTML，以及属性标签页的功能。

Office标签

Office标签页用来添加内容，图形以及分隔线至工具提示。该编辑器将自动地创建工具提示背后全部的HTML代码，以节省您的时间和精力。您在Office标签页下对工具提示所做的任何更改将反映在预览面板中。

Office标签页同样具有一个包含命令按钮的工具栏。您可以通过命令按钮以添加元素至您的工程。









下表描述了Office标签页中的元素：




元素 描述

- 标题** 标题元素出现在工具提示的顶部；默认情况下它显示为粗体。您可以改变此标题的HTML。
- 图像** 该图像将出现在ToolTip正文文本的上方。默认情况下，它会出现在左边。
- 顶部分隔线** 在标题和/或图像之下放置一个水平标尺。该水平标尺为深灰色。
- 正文文本** 正文包含ToolTip的主要内容；它用来提供对指定工具的描述和/或指示。这将出现在标题之后，副标题之前。
- 底部分隔线** 放置一个水平标尺在正文文本下方。该水平标尺颜色为深灰色。
- 副标题** 副标题元素将出现在工具提示的底部。
- Subimage** 该图像显示在工具提示正文文本的下方。默认情况下，它会出现在左边。

下表提供了Office标签页全部命令按钮的描述：

按钮 名称 描述

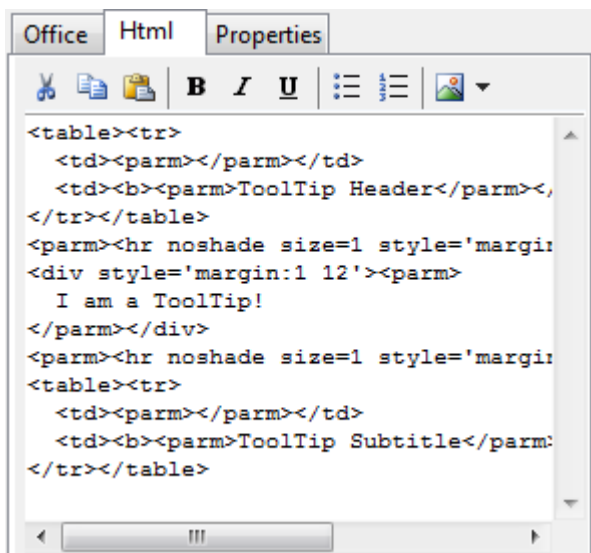
-  剪切 从HTML编辑器窗体移除选中的内容，并将其放置在剪贴板中。
-  复制 从HTML编辑器窗体复制选中的内容至剪贴板。
-  粘贴 将剪贴板中的内容粘贴至HTML编辑器窗体。
-  粗体 向选中的文本添加粗体文本 () HTML标记。
-  斜体 向选中的文本添加斜体文本 (<i></i>) HTML标记。
-  下划线 向选中的文本添加带下划线文本 (<u></u>) HTML标记。

	<p>项目编号列表 添加项目符号列表标签（）至HTML编辑器。</p>
	<p>编号列表 添加编号列表标签（）至HTML编辑器。</p>
	<p>插入图片 通过HTML标签插入一个图像至工具提示（）。</p>




HTML标签

HTML标签页用来向工具提示添加内容或者修改工具提示的内容。当您在HTML标签页创建一个工具提示时，您将具有对工具提示的设计的更多控制。您在HTML标签页上对工具提示所做的任何改变将反映在预览面板中。




HTML标签页由一个带有命令按钮的工具栏以及一个HTML编辑器窗口组成。你可以使用命令按钮添加markup标签到您的工程，或者你可以在HTML编辑器窗口书写代码。



下表提供了HTML标签页上全部的命令按钮的描述：

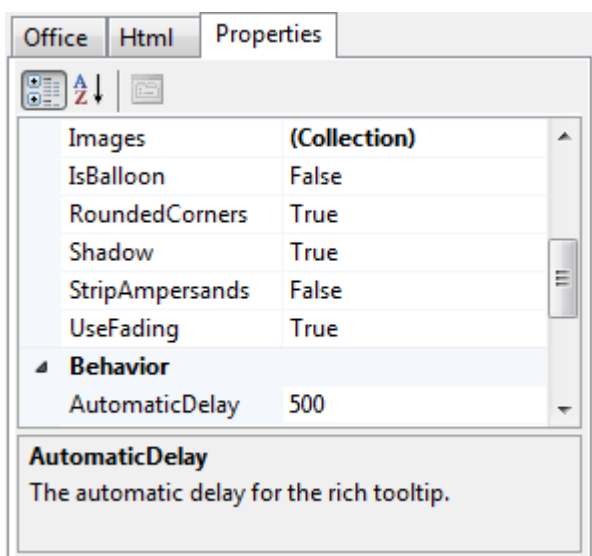
按钮	名称	描述
	剪切	从HTML编辑器窗体移除选中的内容，并将其放置在剪贴板中。
	复制	从HTML编辑器窗体复制选中的内容至剪贴板。
	粘贴	粘贴来自于剪贴板内容至HTML编辑器窗口。
B	粗体	向选中的文本添加粗体文本（）HTML标记。
<i>I</i>	斜体	向选中的文本添加斜体文本（<i></i>）HTML标记。
<u> </u>	下划线	向选中的文本添加带下划线文本（<u></u>）HTML标记。

U

- 
项目符号列表
添加项目符号列表标签（）至HTML编辑器。
- 
编号列表
添加编号列表标签（）至HTML编辑器。
- 
插入图片
通过HTML标签插入一个图像至工具提示（）。

属性标签

属性标签页包含一个PropertyGrid，用于自定义工具提示的外观和行为。您对工具提示所做的大部分改变将反映在预览面板中。



下表列出影响工具提示的外观属性：

属性

描述

- BackColor** 设置工具提示窗口的背景颜色。
- BackgroundImage** 设置提示窗口中显示的背景图像。
- BackgroundImageLayout** 设置背景图像布局为None（默认），Tile，Center，Stretch，Zoom或者TileStretch。
- Border** 设置是否显示工具提示窗口应该显示实线边框。
- BorderColor** 设置工具提示窗口边框的颜色。
- Font** 设置在工具提示使用的默认字体。
- ForeColor** 设置工具提示窗口的前景颜色。
- Images** 获取可以显示在工具提示上的图像的集合。
- IsBalloon** 设置是否显示工具提示为一个气球的形状。如果设置为true，工具提示会像漫画中的对话泡泡。
- RoundedCorners** 设置是否工具提示窗口将具有圆角。
- Shadow** 设置是否提示窗口会显示一个阴影。
- StripAmpersands** 设置是否文本中间的&符号应当显示。
- UseFading** 设置是否在现实一个工具提示时，使用淡出效果。

下表列出影响工具提示行为的属性：

属性	描述
Active	设置一个值，指示是否工具提示当前是活动的状态。
AutomaticDelay	设置工具提示的自动延时。
AutoPopDelay	设置一段时间（单位是毫秒），表示工具提示在鼠标指针悬停不动时，所能够持续显示的时间。
HitTestVisible	指定工具提示是否为鼠标可见。
InitialDelay	设置在工具提示显示之前经过的时间
MaximumWidth	设置工具提示的最大宽度。
Opacity	获取或设置工具提示的不透明度。
ReshowDelay	设置鼠标指针从一个组件移动到另一个组件时，随后出现的工具提示出现的时间必须经过的时间间隔。
ShowAlways	设置一个值，指示是否显示工具提示即使父组件不活动也会显示。

InputPanel 基础

下图展示了当C1InputPanel添加到Winows Form上之后的默认外观:



InputPanel 键盘导航和校验

C1InputPanel 支持方向键导航以及校验功能。

键盘导航

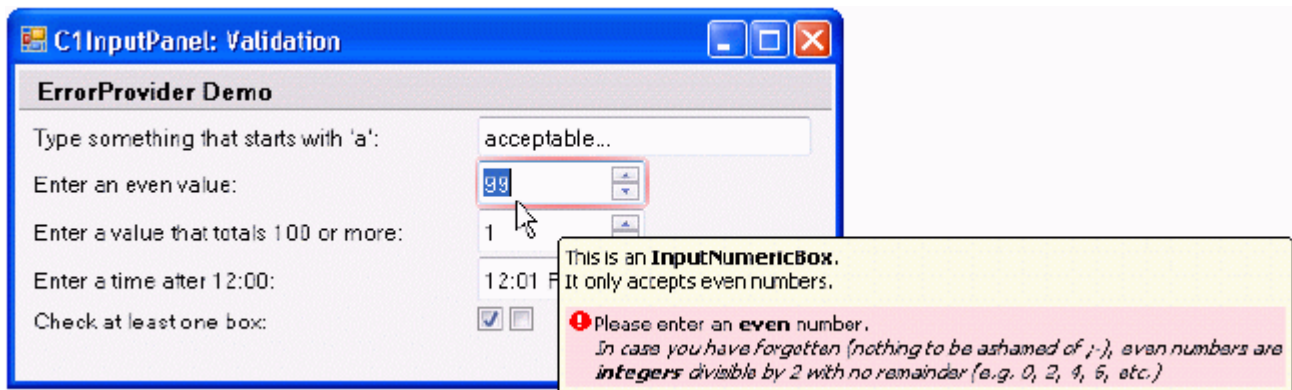
您可以通过按Tab键在C1InputPanel 控件内的各个组件之间导航。如果一个输入字段包含一个自定义的对话框则可以使用F4快捷键打开对话框。如果对话框有几个选项卡窗体,就像字体对话框那样,您可以使用Tab键在每个不同的选项卡窗体之间导航。

校验

当用户输入无效数据时,将出现一个默认红色框,作为一个视觉警告,让用户知道这里存在无效数据。默认红色框的外观可以通过C1InputPanel.ErrorFrameColor以及C1InputPanel.ErrorFrameThickness属性进行自定义。

除了视觉警报,您也可以提供一个表示无效输入的文本警报。C1InputPanel具有一个InputComponent.ErrorText属性,设置该属性将您输入的字符串值作为一个工具提示,当鼠标悬停经过该输入组件时,显示错误信息。

下图显示当使用了该可视化的,以及文本化的校验警报产生的结果:



InputPanel组件

C1InputPanel具有一个Items属性，该属性提供了到InputComponent对象集合的访问。这些对象可以通过设计器，C1InputPanel 项目集合编辑器，或者编程方式进行添加。

本章节提供了组成C1InputPanel控件的一个可视化的描述性的概述。每个主题包含关于元素的一个简短概述以及它如何在C1InputPanel控件中使用。

每个输入组件具有以下主要属性：

- Text
- Enabled
- AutoSize
- Width
- Height
- TabStop
- ToolTipText

下表列出了这些组件并描述其功能：

属性	描述
InputGroupHeader	C1InputPanel中的分组页眉。
InputButton	表示C1InputPanel控件中的一个按钮元素。
InputCheckBox	表示C1InputPanel控件中的一个复选框元素。
InputComboBox	表示C1InputPanel控件中的一个组合框元素。
InputDataNavigator	表示C1InputPanel控件中的一个数据导航元素。
InputDatePicker	表示C1InputPanel控件中的一个日期选择元素。允许用户选择一个日期。
InputImage	C1InputPanel控件上的图像。
InputLabel	C1InputPanel控件中的标签。
InputHtmlLabel	C1InputPanel控件中的HTML标签。
InputMaskedTextBox	表示C1InputPanel控件中的一个掩码输入文本框元素。
InputMenu	表示C1InputPanel控件中的一个菜单元素。
InputNumericBox	表示C1InputPanel控件中的一个数值框元素。
InputProgressBar	表示C1InputPanel控件中的一个进度条元素。
InputRadioButton	表示C1InputPanel控件中的一个单选按钮元素。
InputSeparator	表示C1InputPanel控件中的一个分隔线元素。
InputSplitButton	表示C1InputPanel控件中的一个分隔按钮元素。
InputTextBox	表示C1InputPanel控件中的一个文本框元素。
InputTimePicker	表示C1InputPanel控件中的一个时间选择器元素。
InputTrackBar	表示C1InputPanel控件中的一个滑块元素。
InputControlHost	表示C1InputPanel控件中的一个控件宿主元素。

InputButton

InputButton组件，和标准的按钮控件类型一样，允许用户单击以执行某个动作。



除了标准的按钮属性之外，InputButton包含以下独特的成员：

- `InputComponent.Break`属性 - 获取或设置流布局如何在组件之后发生打断。
- `InputButton.CheckOnClick` 属性 - 获取或设置一个值，该值表示是否当按钮单击之后切换为始终按下的状态。
- `InputButton.Pressed` 属性 - 获取一个值，该值表示是否按钮为按下状态。
- `InputComponent.ToolTipText` 属性 - 获取或设置一个文本值，该值将出现在该组件的工具提示上。
- `InputButton.PressedChanged` 事件 - 当`Pressed`属性的值发生变化时触发。

InputCheckBox

您可以设置该复选框的对齐方式为靠其文本的左侧，居中或者右侧对齐。该 `InputCheckBox` 可以具有三种不同的状态：

- `Unchecked`
- `Checked`
- `Indeterminate`

默认情况下，`InputCheckBox`允许两种复选状态，`Unchecked`以及`Checked`，但是您也可以通过设置 `InputCheckBox.ThreeState`属性的值为`True`允许出现全部的三种复选状态。

您可以通过`InputCheckBox.CheckAlign`属性指定复选框出现在左侧，居中或者右侧对齐。下表说明了当 `InputCheckBox`组件的`InputCheckBox.CheckAlign`属性设置为不同值时的不同表现：

InputCheckBox.CheckAlign 外观设置

Left

 A screenshot of a checkbox control where the text "CheckBox" is aligned to the left of the checkbox box.

Center


 A screenshot of a checkbox control where the text "CheckBox" is centered horizontally relative to the checkbox box.

Right

 A screenshot of a checkbox control where the text "CheckBox" is aligned to the right of the checkbox box.

InputRadioButton

`InputRadioButton`是一种按钮，当单击时，重置相同分组中的其他按钮的选中状态。

 Radio Button

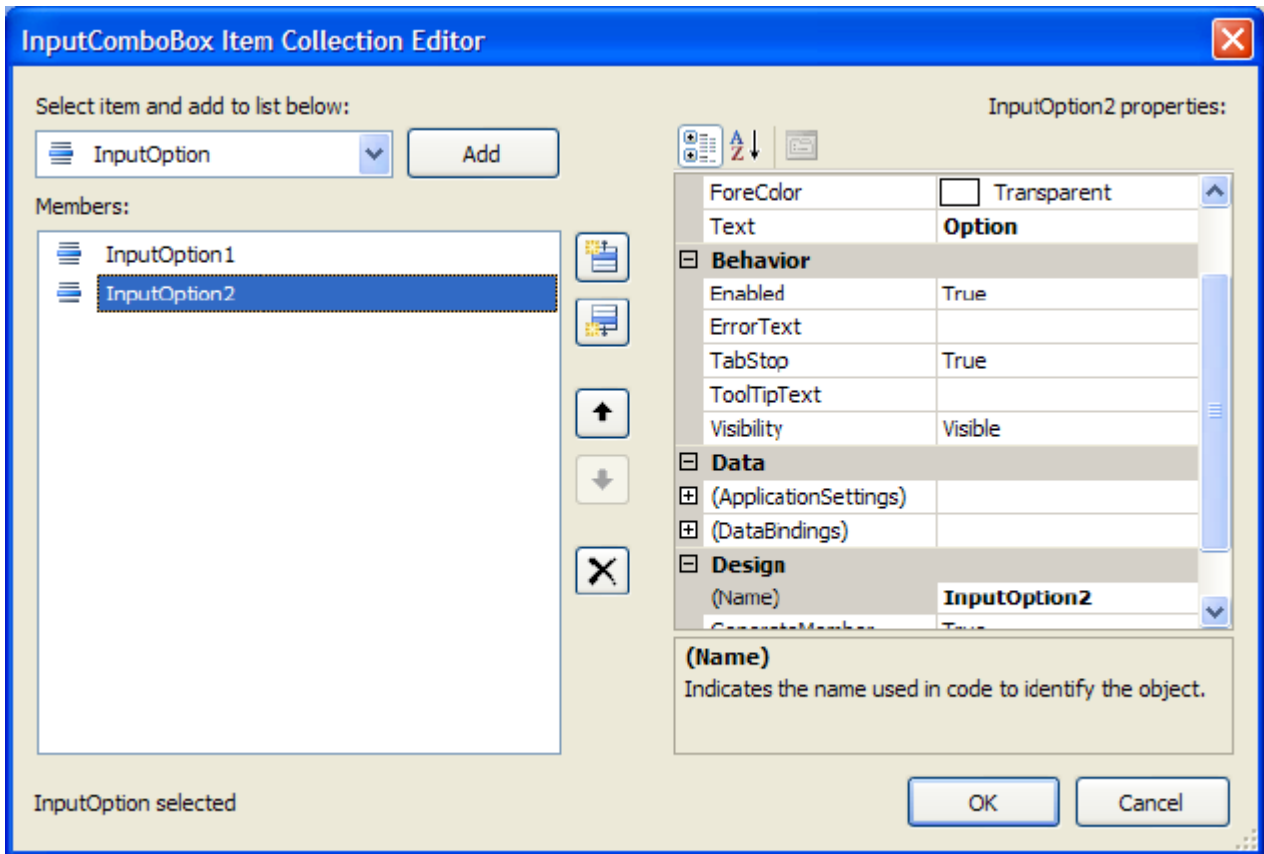
InputComboBox

默认的`InputComboBox`具有一个下拉箭头，但是您可以通过设置其`DropDownStyle`属性改变下拉样式的外观。`InputComboBox`具有一个下拉箭头以及一个下拉列表框。

您可以使用`InputComboBox.MaxDropDownItems`属性指定显示在下拉列表框中的项目的最大个数。默认值是显示八个项目。

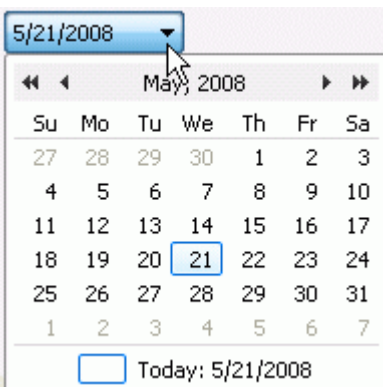
有两种类型的项目可以添加到`InputComboBox`：`InputOption`以及`InputGroupHeader`。

`C1InputPanel`提供了一个`InputComboBox`项目集合编辑器，以便向`InputComboBox`添加项目。添加的项目为 `InputOptions`类型。



InputDatePicker

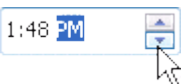
在运行时InputDatePicker的外观如下:



通过单击方框中的下拉箭头，将出现一个在底部显示当前日期的日历。您可以通过设置InputDatePicker.ShowToday属性为False，以选择不在日历底部显示当前日期。

InputTimePicker

在运行时InputTimePicker组件外观如下所示:



InputTimePicker包含三个字段以输入时间。默认的InputTimePicker包含以下三个字段：小时，分钟，以及上午/下午。

InputGroupHeader

该组件内嵌在C1InputPanel的主控件中。它为内嵌的组件显示一个边框，同时具有一个可折叠的标题栏。可以通过InputGroupHeader.Text属性设置标题栏的文本。默认文本，Group，将出现在页眉的标题上。

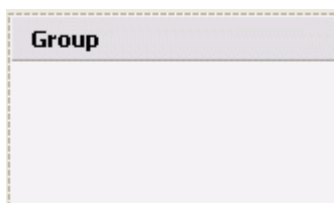
InputGroupHeader组件可以折叠收起，当InputGroupHeader.Collapsible属性设置为True时。更多信息，请参见创建一个可折叠的输入分组。



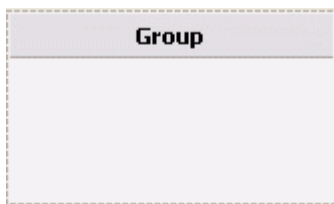
如果您希望InputGroupHeader显示为收起状态，您可以设置InputGroupHeader.Collapsed属性的值为True。InputGroupHeader的文本可以相对于分组靠左，居中或者靠右对齐，通过设置InputGroupHeader.TextAlign属性。

TextAlign 外观 Property 设置

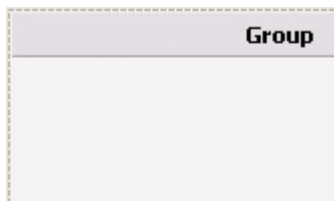
Left



Center



Right



InputImage

InputImage组件用作在C1InputPanel中间显示图像。您可以通过使用InputImage.SizeMode属性决定如何决定在组件内部显示的图像的尺寸。



InputLabel

InputLabel组件提供描述性的文本，以提示用户关于该表单的目的。

当您指定了InputLabel的宽度时，标签将折行显示文本，并且其高度将自动调整以适应文本的高度。



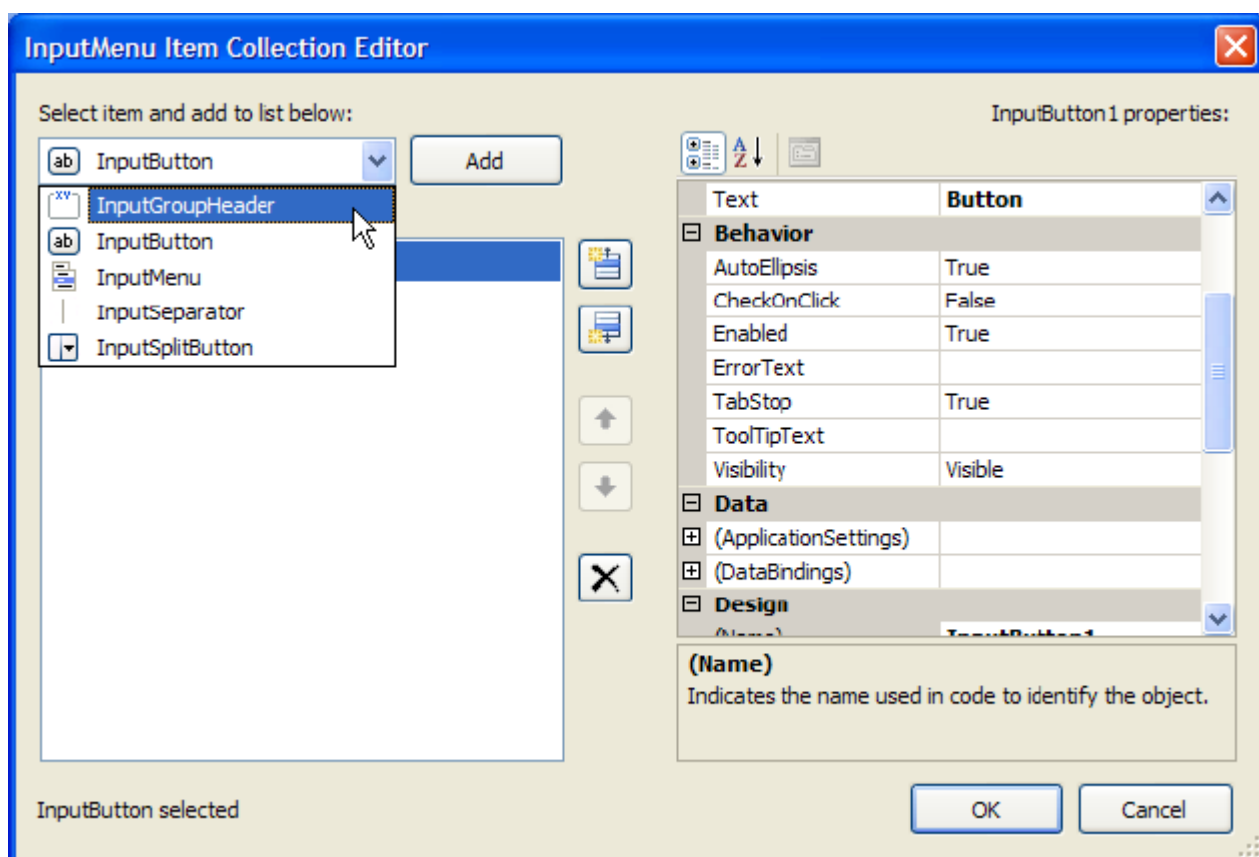
InputHtmlLabel

InputHtmlLabel是一个HTML Label组件，用作提供描述性的文本以提示用户关于该表单的目的。



InputMenu

InputMenu 组件是一个菜单。您可以很容易地通过InputMenu项目集合编辑器添加菜单和菜单项：



InputProgressBar

InputProgressBar是一个用来显示某一个任务完成程度的进度条，例如，当您下载一个文件时，显示下载文件的进度。

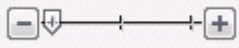
当您向C1InputPanel添加一个InputProgressBar时，看起来如下图所示：



InputTrackBar

InputTrackBar表示C1InputPanel控件中的一个滑块元素。

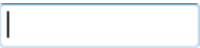
当您向C1InputPanel添加一个InputTrackBar组件时，看起来如下图所示：



在运行时，您可以向左移动箭头以减少值或者向右移动箭头以增加绑定对象的值。您可以通过设置InputTrackBar.TickFrequency属性改变每一次步进改变值的大小。

InputMaskedTextBox

InputMaskedTextBox组件用做将输入的文本以掩码方式表示。组件中的每一个字符位置映射到一个特殊的占位符或者文本字符。文本字符，或者文本，可以针对所使用的数据类型给出提示。例如，括起电话区号数字的括号以及破折号属于文本字符。(412)-123-2222。编辑掩码可以防止用户向组件输入无效字符，并对用户界面提供了其他增强。



InputMaskedTextBox和标准的MaskedTextBox相似，并且和ComponentOne ASP.NET 版本的WebInput中的C1WebMaskEdit控件几乎完全一致，除了以下的独特的成员：

- InputComponent.Break
- InputComponent.BoundValueChanged

InputNumericBox

在运行时，InputNumericBox组件外观和下图类似：



InputNumericBox包含一个用于输入数值的字段以及右侧的上下箭头。单击向上或者向下箭头一次将当前值增加/减少值1，除非您指定了一个不是1的值给InputNumericBox.Increment属性的值。您可以通过指定Font以及ForeColor属性自定义数值的外观。同样您也可以通过设置InputTextBox.TextAlign属性将数值靠左，居中，或者靠右对齐。

Input TextBox

InputTextBox组件在运行时外观如下图所示：



InputTextBox组件用作输入文本。您可以通过修改以下任意的行为相关的属性修改该行为：

- InputTextBox.AcceptsReturn –获取或设置一个值，该值决定是否InputTextBox应当处理Enter键。
- InputTextBox.AcceptsTab –获取或设置一个值，该值决定是否InputTextBox应当处理Tab键。
- InputComponent.Enabled –获取或设置一个值，该值指示是否该组件可以响应用户交互。
- InputComponent.ErrorText –获取或设置一个错误描述字符串，关联到该组件。
- InputTextBox.MaxLength –获取或设置该组件允许输入的最大字符数。
- InputTextBox.Multiline –获取或设置一个值，该值决定是否用户可以输入多行文本至InputTextBox。
- InputComponent.TabStop –获取或设置一个值，该值指示是否用户可以通过TAB键将焦点移动到该组件上。
- InputComponent.ToolTipText –获取或设置作为一个组件工具提示出现的文本。
- InputComponent.Visibility –获取或设置该组件是否可见。
- InputTextBox.WordWrap –获取或设置是否文本允许在InputTextBox内部折行显示。

您可以通过修改以下任意的外观属性自定义InputTextBox组件的外观：

- InputComponent.Font –获取或设置该InputComponent使用的字体。
- InputComponent.FontPadding –获取或设置是否InputComponent应当使用padding以防止文本被剪裁。
- InputComponent.ForeColor –获取或设置用作显示InputComponent文本的颜色。
- InputTextBox.Text –获取或设置关联到此组件的文本。
- InputTextBox.TextAlign –获取或设置文本如何在InputTextBox中进行对齐。

当您具有跨越多行的文本时，您需要确保设置了InputTextBox.Multiline属性的值为True。

InputDataNavigator

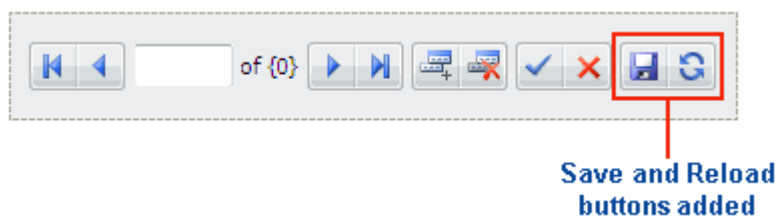
这是一个提供方便的在数据源记录之间导航按钮的数据绑定组件。它允许移动到第一个，最后一个，前一个或者后一个记录，并且通用的数据操作，例如更新数据远以及刷新数据等。

当您添加InputDataNavigator至C1InputPanel之后，看起来类似下图：



您可以通过设置NavigatorItems属性选择哪些项目在InputDataNavigator上可见。关于如何选择InputDataNavigator按钮的更多信息，请参见为InputDataNavigator选择项目。

您还可以添加一个Save按钮以及一个Reload按钮至InputDataNavigator组件（参见添加Reload，Save以及Edit按钮）下图描绘了添加了Save以及Reload按钮的InputDataNavigator组件的外观：



下表提供了关于InputDataNavigator组件按钮的相关信息：

图 像	名称	描述
	Move First	移动到第一个记录。
	Move Previous	移动到前一个记录。
	Move Next	移动到下一个记录。
	Move Last	移到最后一条记录。
	Add New	添加一条新的记录。
	Delete	删除当前记录。
	Apply Changes	将更改应用于当前记录。一旦使用了Apply Changes按钮，则发生的更改不能够撤销。
	Cancel Changes	取消对当前记录的修改。如果已经按下了Apply Changes按钮将变更应用到记录，则无法撤销更改。
	Save Data	保存当前数据至外部数据源。
	Refresh Data	从外部数据远获取更新后的数据。

InputControlHost

InputControlHost使得可以嵌入任意的控件至C1InputPanel。完成以下步骤以添加一个MonthCalendar控件至输入面板:

1. 向窗体添加C1InputPanel控件。
2. 我们必须定义以下类:

Visual Basic

Visual Basic

```
Public Class MonthCalendarHost
    Inherits C1.Win.C1InputPanel.InputControlHost
    Public Sub New()
        MyBase.New(New System.Windows.Forms.MonthCalendar())
    End Sub
End Class
```

C#

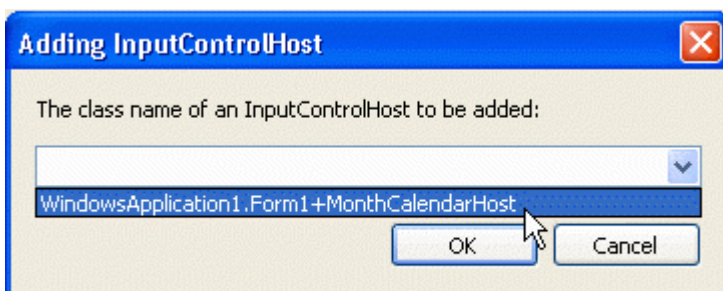
C#

```
public class MonthCalendarHost : C1.Win.C1InputPanel.InputControlHost
{
    public MonthCalendarHost()
        : base(new System.Windows.Forms.MonthCalendar())
    {
    }
}
```

3. 编译该工程。
4. 在设计视图中, 选择C1InputPanel控件, 并打开其智能标签, 接下来在“Add New Item”组合框中选择“InputControlHost”。

这将打开一个对话框, 在此您应当输入我们的控件宿主的类名称, 比如说“WindowsApplication1.MonthCalendarHost”。

5. 在Adding InputControlHost对话框中单击OK。该控件将添加到C1InputPanel。
下次, 当您希望添加MonthCalendarHost组件时, 您将不需要键入其名称。该类型的名称可以从“Adding InputControlHost”对话框中的下拉列表中选择。



注: 强烈建议您将全部的基于InputControlHost的组件移动到独立的.DLL工程, 这将使得这些组件不会和主程序一起重新构建。这可以让您避免以下众所周知的问题: 在VS2008IDE中, 当应用程序进行过一次调试之后, 窗体的任何属性或者包含一个InputControlHost的C1InputPanel无法被修改。

InputPanel 组件布局

默认情况下，Input组件按照每行一个的方式堆叠显示。您可以改变输入组件的布局流，通过输入元素的InputComponent.Break属性。InputComponent.Break属性具有以下可选的四种设置：

- None – 布局将不会在该组件之后打断流布局。
- Row – 默认设置。设置每一个组件在流布局中放置一个新行。
- Column – 在组件之后创建一个新列。
- Group – 在当前输入组件下方创建一个新的起始行，位于当前分组的左侧边缘。

C1InputPanel 提供了两种不同的方式来指定输入组件布局的打断位置：

- 如果第一个组件的Break 属性设置为None

或

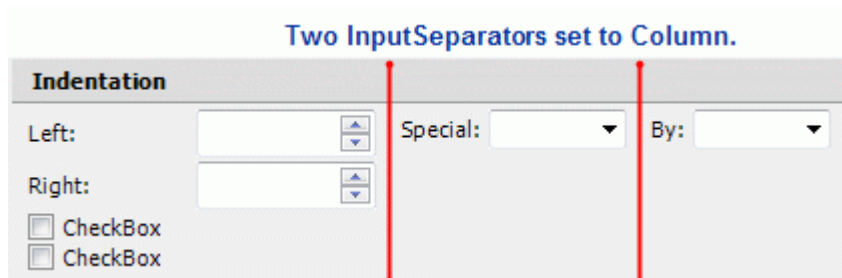
- 如果输入组件之间存在一个InputSeparator

在InputGroup组件内部的子项目缩进

C1InputPanel同时也处理位于InputGroupHeader内部元素的布局。InputGroupHeader组件具有一个带有内边距的内容区域，因此内部的元素将自动具有缩进。

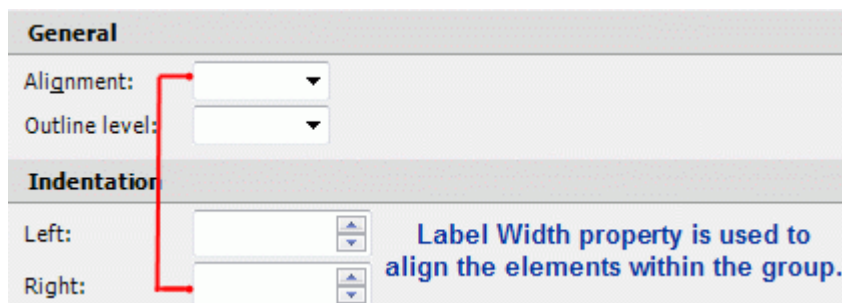
多列

通过InputSeparator 组件，可以在一个分组中创建多列。默认情况下，当InputSeparator 添加到C1InputPanel控件上时，它是不可见的，但是可以使得其可见以创建垂直或者水平的线，以便将控件分隔成多行或者多列。下图说明了InputSeparator 是如何将Label, Special, 以及其他Label分隔成多列的：



在分组内对齐元素

此外，对于InputLabel当指定了其标签宽度时，标签将折行，同时其高度将适应文本。



InputPanel for WinForms 示例

请注意，这个ComponentOne软件工具伴随着各种示例工程和/或Demo，可能同时使用了包含在ComponentOne套件包中的其他开发工具。

示例可以通过ComponentOne示例资源管理器进行访问。为查看示例，在您的桌面上，单击开始按钮，并单击ComponentOne | Studio for WinForms | Samples | InputPanel Samples。下表为每一个示例提供了一个简短的描述。

Visual Basic示例

示例	描述
CustomEditors	演示用作编辑C1InputPanel视觉样式的自定义编辑器。
DataBinding	演示如何使用特殊属性将每个类型的输入组件绑定到数据源。
ErrorProvider	演示如何使用组件的验证事件对其进行验证。它同时演示了如何使用ErrorText属性，当用户在输入组件存在非法输入时，创建一个错误提示信息。

C#示例

示例	描述
CustomEditors	演示用作编辑C1InputPanel视觉样式的自定义编辑器。
DataBinding	演示如何使用特殊属性将每个类型的输入组件绑定到数据源。
ErrorProvider	演示如何使用组件的验证事件对其进行验证。它同时演示了如何使用ErrorText属性，当用户在输入组件存在非法输入时，创建一个错误提示信息。