

第5章

VB.NET 控件技术

VB.NET 窗体上可以使用的控件种类非常多，不同类型的控件之间既有共通之处，也有控件自身的个性和特点。

大部分控件的功能是用来呈现或存储程序中的数据，本章按照控件的复杂程度、功能的划分，分类讲解比较常用的控件。

一般的控件，在设计期间和运行期间会占据窗体上的一个区域，因此控件都具有位置和大小这些常规属性。

但有一部分控件，运行期间不占据窗体，需要的时候才弹出来。例如右键菜单、文件打开和保存对话框、计时器等，这些可以认为是“隐形控件”。

本章要点：

- ❑ 掌握 RichTextbox、CheckBox、ListBox、Timer 这几个使用频率最高的控件的用法。
- ❑ 了解 ListView、Treeview 控件添加条目的方法。

5.1 文本编辑类控件

这类控件主要包括 TextBox 和 RichTextBox。

TextBox 适用于输入用户名、密码等单行信息，RichTextBox 适用于装载 rtf 格式的富文本内容。

5.1.1 TextBox

文本框控件默认是单行模式，即使内容是多行的也显示为单行。MultiLine 属性设置为 True 即可成为多行文本框。

- ❑ 为文本框设置文本的属性有：Text、AppendText，分别用来设置文本、追加文本。
- ❑ 文本框的 Lines 属性允许把一个数组的内容一次性赋给文本框。

❑ 文本框中所选文字部分有关的属性有：SelectionStart、SelectionLength。

❑ 文本框内容发生修改触发的事件是 TextChanged 事件。

下面的程序，把一个数组的内容直接赋给文本框，并且自动选中第 2 个字符位置，连续选中 4 个字符。

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    Dim poetry As String() = New String(3) {"春雨惊春清谷天", "夏满芒夏暑相连", "秋处露秋寒霜降", "冬雪雪冬小大寒"}
    With Me.TextBox1
        .Multiline = True
        .Size = New Point(x:=200, y:=100)
        .Lines = poetry
        .SelectionStart = 2
        .SelectionLength = 4
        .Focus()
    End With
End Sub
```

启动窗体，单击按钮，文本框中自动输入了内容，并且自动选中一部分文字，如图 5-1 所示。

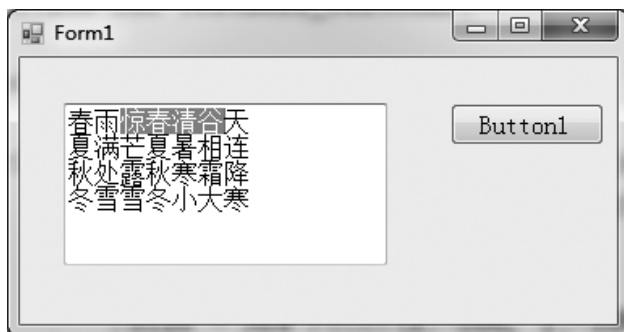


图 5-1 自动设置文本框

■ 5.1.2 RichTextBox

RichTextBox 是富文本框控件，与 TextBox 最大的不同是，该控件可以像 Word 文档一样保存文字的格式、图片等对象。TextBox 控件的所有文字的格式（字体名称、大小等）都是统一的，而 RichTextBox 则可以对部分文字进行格式设定。

前面所述的 TextBox 控件具有的属性、事件，RichTextBox 都具备。

RichText 控件对所选部分进行格式设定的常用属性如下。

- ❑ SelectionFont：所选部分的字体。
- ❑ SelectionColor：所选部分的颜色。
- ❑ SelectionBackColor：所选部分的背景色。



此外, RichTextBox 控件还可以使用 LoadFile 方法载入文本文件或 rtf 文档的内容, 也可以把控件中的内容使用 SaveFile 方法保存到文件中。

下面的程序自动选中一部分文字, 并设置所选部分的格式。

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    With Me.RichTextBox1
        .Text = " 春雨惊春清谷天 "
        .SelectionStart = 2
        .SelectionLength = 3
        MsgBox(" 所选文字是: " & .SelectedText)
        .SelectionBackColor = Color.Yellow
        .SelectionFont = New Font(" 微软雅黑 ", 20, FontStyle.Italic Or FontStyle.Bold)
        .SelectionColor = Color.Blue
        .Focus()
    End With
End Sub
```

启动窗体, 单击按钮, RichTextBox 控件中自动选中 3 个字, 如图 5-2 所示。



图 5-2 设置选中部分的字体样式

也可以在 RichTextBox 控件中编辑图片等非文字对象。

■ 5.1.3 MaskedTextBox

MaskedTextBox 是掩码文本框控件, 需要事先设置 Mask 属性, 从而让用户按照掩码规定的格式进行输入。

在该控件的属性窗口中, 找到 Mask 属性, 弹出“输入掩码”的设置对话框, 如图 5-3 所示。



图 5-3 掩码设置

例如，设置为 18 位身份证号码的掩码：

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    MsgBox(" 掩码是: " & Me.MaskedTextBox1.Mask & vbNewLine & " 文本是: " & Me.MaskedTextBox1.Text)
End Sub
```

启动窗体，输入一部分号码，尚未输入的部分显示为下画线。

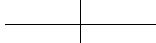
单击按钮，对话框中弹出掩码以及文本框中目前的文本，如图 5-4 所示。



图 5-4 掩码文本框结果

5.2 标签类控件

标签类控件有 Label、LinkLabel。LinkLabel 控件用于在窗体上显示超链接。



5.2.1 Label

Label 是最常用的标签控件，AutoSize 属性默认为 True，也就是控件的大小会自动适应文本的长度。默认不显示边框线，并且文本默认对齐到控件的左上角。

为了便于说明，窗体上放置一个 Label 控件，设置其 AutoSize 属性为 False，也就是无论容纳多少文字，控件的大小不会自动改变。

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    With Me.Label1
        .BorderStyle = BorderStyle.FixedSingle
        .TextAlign = ContentAlignment.MiddleCenter
        .Text = "VSTO 开发 " & vbCrLf & " 中级教程 "
    End With
End Sub
```

启动窗体，单击按钮控件，标签控件自动显示为具有边框线，而且文本在水平、垂直方向都居中，如图 5-5 所示。



图 5-5 设置标签控件的对齐方式

5.2.2 LinkLabel

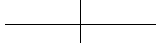
LinkLabel 控件的功能和外观类似于网页中的超链接。

下面的程序，单击按钮会自动设置 LinkLabel 控件的显示文本，以及超链接的网址。

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    With Me.LinkLabel1
        .Text = " 刘永富的博客园 "
        .Links.Item(0).LinkData = "https://www.cnblogs.com/ryueifu-VBA"
    End With
End Sub
```

为了实现单击 LinkLabel 就用默认浏览器打开网页，还需要设置该控件的 LinkClicked 事件。

```
Private Sub LinkLabel1_LinkClicked(sender As Object, e As LinkLabelLinkClickedEventArgs) Handles LinkLabel1.LinkClicked
    System.Diagnostics.Process.Start(e.Link.LinkData.ToString)
End Sub
```



代码分析: System.Diagnostics.Process.Start 类似于 VBA 中的 Shell, 可以启动指定的应用程序, 例如 System.Diagnostics.Process.Start(fileName:="Notepad.exe", arguments:="E:\VSTO\日志.txt") 可以自动启动记事本打开一个文件。

启动窗体, 单击按钮, LinkLabel 控件的外观如图 5-6 所示。

然后单击 LinkLabel 标签, 在默认浏览器中自动打开博客园的页面。

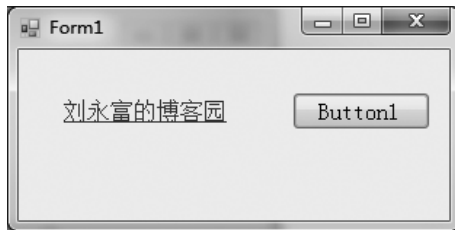


图 5-6 窗体上的超链接

5.3 选择类控件

选择类控件有 CheckBox、RadioButton。

5.3.1 CheckBox

复选框控件有两个属性来获取、设置控件的勾选状态。

❑ Checked: 是否勾选, True 或 False 两种情形。

❑ CheckState: 勾选状态, 有勾选、非勾选、中间状态 3 种情形。

复选框控件的重要事件如下。

❑ CheckChanged: 被勾选、取消勾选时发生。

❑ CheckStateChanged: 勾选状态发生变化时发生。

窗体上放置 3 个 CheckBox 控件, 单击按钮自动勾选、取消勾选复选框。

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    Me.CheckBox1.Checked = True ' 等价于 Me.CheckBox1.CheckState = CheckState.Checked
    Me.CheckBox2.Checked = False ' 等价于 Me.CheckBox2.CheckState = CheckState.Unchecked
    Me.CheckBox3.CheckState = CheckState.Indeterminate
End Sub
```

启动窗体, 单击按钮, 各个复选框的外观如图 5-7 所示。

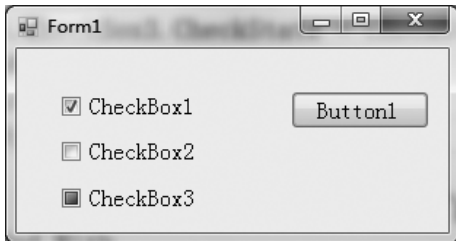
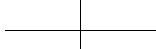


图 5-7 自动设置复选框



手工不能把复选框设置为中间状态。

5.3.2 RadioButton

RadioButton 是单选按钮，在多个同类控件中只能有一个处于选中状态。由于各个单选按钮存在的互斥性，同一个窗体上需要多组单选按钮的情形下，就需要使用不同的 GroupBox 控件来分隔。GroupBox 是框架控件，用来容纳其他控件，便于统一管理。

RadioButton 控件的 Checked 属性用来读写控件的选中状态，当单选按钮发生选中状态的变化时触发 CheckedChanged 事件。

窗体上放置两个 GroupBox 控件，然后放置一些 RadioButton，通过单击按钮自动勾选单选按钮。

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    Me.RadioButton2.Checked = True
    Me.RadioButton5.Checked = True
End Sub
```

启动窗体，单击按钮，第 2 个和第 5 个单选按钮被自动勾选，如图 5-8 所示。



图 5-8 自动单击单选按钮

5.4 列表条目类控件

列表条目类控件有 ComboBox、ListBox、CheckedListBox。

5.4.1 ComboBox

ComboBox 是组合框控件，为组合框添加条目的方法如下。

- ❑ Items.Add: 添加一个条目。
- ❑ Items.AddRange: 批量添加数组中的各个元素。
- ❑ DataSource: 把数组等集合对象作为组合框的数据源。

❑ **Items.Clear**: 清空组合框中所有条目。

组合框控件不仅可以单击下拉框选择条目, 该控件还兼具 **TextBox** 控件的功能, 可以直接输入新的内容。

下面的程序, 向组合框中添加 3 个条目, 然后设置 **Text** 属性。

```
Private Sub ComboBox()
    With Me.ComboBox1
        .Items.Clear()
        .Items.AddRange({"Excel", "PowerPoint", "Word"})
        .Text = "Outlook"
        MsgBox(" 条目总数 " & .Items.Count)
        .Focus()
    End With
End Sub
```

启动窗体, 组合框中的文本是 “ Outlook”, 但是这个并不是一个条目, 如图 5-9 所示。

组合框控件的 **SelectedIndex** 属性表示所选条目的索引, 例如 **ComboBox1.SelectedIndex = 2**, 表示自动选中第 3 个条目。条目被选中时触发 **SelectedIndexChanged** 事件。

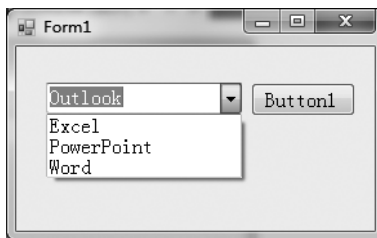


图 5-9 组合框控件

■ 5.4.2 ListBox

ListBox 是列表框控件, 用于把多个条目显示为列表。

为列表框添加条目的方式与 **ComBoBox** 相同。

列表框的重要属性如下。

- ❑ **MultiColumn**: 该属性默认值为 **False**, 如果设置为 **True**, 则是多列显示。
- ❑ **SelectionMode**: 选择模式, 默认是单选模式。
- ❑ **SelectedIndex**: 选中的条目索引, 当没有任何条目被选中时, 该属性为 **-1**。
- ❑ **SelectedItems**: 多选模式下被选中的所有条目。
- ❑ **GetSelected**: 返回指定索引的条目是否被选中。
- ❑ **Text**: 当前选中条目的文本。

列表框的重要事件如下。

SelectedIndexChanged: 当列表框选中条目时触发。

下面的程序, 自动向列表框控件中添加若干条目, 然后自动选中某一条目。

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    With Me.ListBox1
        .Items.Clear()
        .Items.Add(" 星期一 ")
    End With
End Sub
```



```
.Items.Add(" 星期二 ")
.Items.Add(" 星期三 ")
.Items.AddRange({" 星期四 ", " 星期五 ", " 星期六 ", " 星期日 "})
.SelectionMode = SelectionMode.One           ' 单选模式
.SelectedIndex = 3                           ' 选中第 3 条目
.Focus()

End With
End Sub
```

代码分析：列表框控件默认样式是单选模式、单列模式。当选中新的条目，之前选中的会自动取消选中。

启动窗体，单击按钮，列表框的第 4 个条目处于选中状态（最上面条目的索引是 0），如图 5-10 所示。

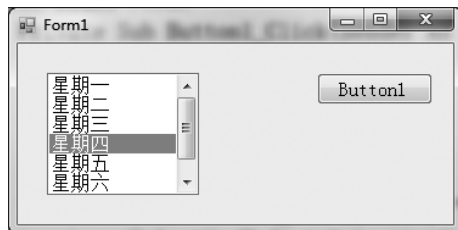


图 5-10 列表框控件

把 ListBox 控件的 SelectionMode 设置为 MultiExtended 时，为多选模式，可以按 Ctrl 键或 Shift 键选择多个条目，如果要了解被选中的有哪些条目，可以遍历列表框的所有条目，然后把 GetSelected 返回 True 的条目单独列出。

```
Private Sub 多选模式 ()
    With Me.ListBox1
        .Items.Clear()
        .Items.AddRange({" 星期一 ", " 星期二 ", " 星期三 ", " 星期四 ", " 星期五 ", " 星期六 ", " 星期日 "})
        .SelectionMode = SelectionMode.MultiExtended ' 多选模式
        .SetSelected(index:=2, value:=True)         ' 第 2 条处于选中
        .SetSelected(index:=3, value:=False)        ' 第 3 条处于未选中
        .SetSelected(index:=4, value:=True)         ' 第 4 条处于选中
        For i = 0 To .Items.Count - 1
            If .GetSelected(index:=i) Then
                MsgBox(.Items.Item(i))
            End If
        Next i
        .Focus()
    End With
End Sub
```

启动窗体，单击按钮，列表框自动选中“星期三”“星期五”，然后对话框中显示所选条目的文本，如图 5-11 所示。

列表框的 MultiColumn 属性设置为 True 时，为多列模式。单列模式时，条目太多时列表框自动显示垂直滚动条。多列模式时，条目太多时会自动甩到右侧的列中。ColumnWidth 可以设置列宽。

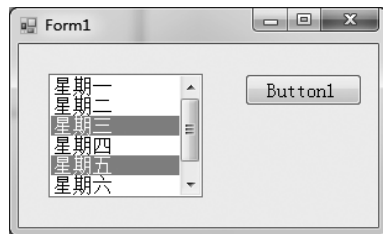


图 5-11 多选模式的列表框



下面的程序，列表框设置为多列模式。

```
Private Sub 多列模式 ()  
    With Me.ListBox1  
        .MultiColumn = True  
        .ColumnWidth = 80  
        .DataSource = {"January", "February", "March", "April", "May", "June",  
"July", "August", "September", "October", "November", "December"}  
    End With  
End Sub
```

启动窗体，单击按钮，列表框的高度只能显示 7 个条目，因此从第 8 个条目显示在右侧，如图 5-12 所示。

列表框在多列模式下，其属性、方法、事件与单列模式完全一样，只是显示的外观不同而已。

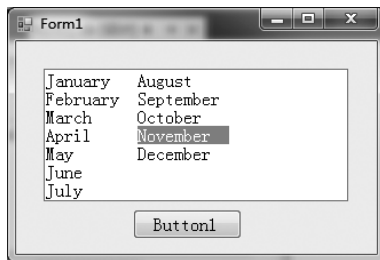


图 5-12 多列模式的列表框

5.4.3 CheckedListBox

CheckedListBox 控件是带有复选框的列表框控件，因此必须理解勾选（Check）和选择（Select）这两种行为。勾选是指单击条目左侧的方框，而选择是指单击条目。

该控件的对象成员与 CheckBox 控件、ListBox 控件均有共同之处。

风格样式方面的重要属性有 MultiColumn、SelectionMode 等，但是 CheckedListBox 不支持多重选择。

为 CheckedListBox 控件添加条目的方法和 ComboBox、ListBox 一样。

由于 CheckedListBox 控件只能选择一个条目，所以 SelectedIndex 属性用来获取、设置所选条目的索引。

CheckedListBox 控件可以勾选多个条目，使用 GetItemChecked 判断指定条目是否被勾选，GetItemCheckState 返回条目的勾选状态。

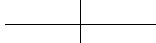
CheckedListBox 控件的常用事件如下。

❑ ItemCheck：勾选 / 取消勾选时触发。

❑ SelectedIndexChanged：选择条目时触发。

下面的程序为 CheckedListBox 控件自动添加条目，然后设置为多列模式。

```
Private Sub CheckedListBox()  
    With Me.CheckedListBox1  
        .Items.Clear()  
        .Items.AddRange({"语文", "数学", "英语", "政治", "历史", "物理", "化学",  
"地理", "生物", "体育"})  
        .MultiColumn = True  
        .ColumnWidth = 60  
        .SetSelected(index:=2, value:=True)  
    End With  
End Sub
```



```
.SetItemChecked(index:=3, value:=True) ' 勾选第 3 个条目  
.SetItemChecked(index:=5, value:=True) ' 勾选第 5 个条目  
.SetItemCheckState(index:=7, value:=CheckState.Indeterminate)  
                                ' 设置第 7 个条目的勾选状态为中间状态  
  
End With  
End Sub
```

启动窗体，单击按钮，会看到英语被选中，政治、物理被勾选，地理处于中间状态，如图 5-13 所示。

当控件中有多个条目被勾选时如何遍历勾选了的条目呢？可以遍历复选列表框的所有条目，然后利用 `GetItemCheckState(i)` 来判断第 *i* 个条目的勾选状态。

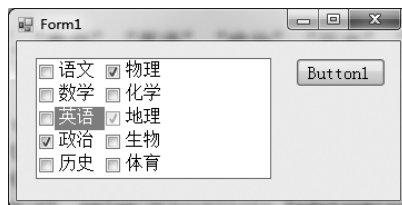


图 5-13 CheckedListBox 控件

```
For i As Integer = 0 To .Items.Count - 1  
    If .GetItemCheckState(index:=i) = CheckState.Checked Then  
        MsgBox(.Items.Item(i))  
    End If  
Next i
```

运行上述程序，对话框依次弹出“政治”和“物理”。

5.5 数值调节类控件

数值调节类控件有滚动条、微调按钮等。

5.5.1 HScrollBar 和 VScrollBar

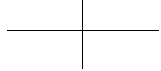
HScrollBar、VScrollBar 是水平、垂直滚动条控件。

该控件的 `Minimum`、`Maximum` 属性获取和设置控件的最小值和最大值，`Value` 属性获取和设置滚动条的数值。

滚动条控件的数值发生变化时，触发 `ValueChanged` 事件。

窗体上分别放置一个水平、垂直滚动条控件，运行下面的程序自动设置。

```
Private Sub HScrollBar()  
    With Me.HScrollBar1  
        .Minimum = 60  
        .Maximum = 100  
        .Value = 85  
    End With  
    With Me.VScrollBar1  
        .Minimum = 60  
        .Maximum = 100  
        .Value = 65  
    End With  
End Sub
```



启动窗体，单击按钮，自动更改滚动条的数值，如图 5-14 所示。

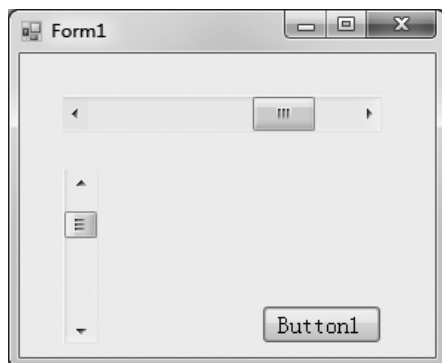


图 5-14 滚动条控件

■ 5.5.2 TrackBar

TrackBar 是滑块控件，有点像天平上的砝码。

其属性和事件与滚动条控件大同小异。

TickStyle 属性可以设置刻度的位置，默认在滑块的下方显示刻度。

下面的程序，自动设置滑块控件。

```
Private Sub TrackBar()  
    With Me.TrackBar1  
        .Minimum = 60  
        .Maximum = 100  
        .Value = 85  
    End With  
End Sub
```

启动窗体，单击按钮，滑块控件的结果如图 5-15 所示。

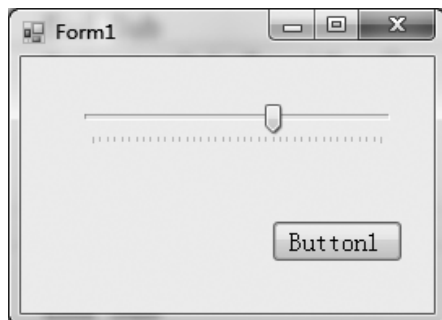


图 5-15 滑块控件

■ 5.5.3 NumericUpDown

NumericUpDown 控件是数值调节器。



该控件默认的最小值、最大值、步长都是整数，而且该控件默认只显示整数部分。

下面的程序，使用数值调节器控件进行小数的调节。

```
Private Sub NumericUpDown()  
    With Me.NumericUpDown1  
        .DecimalPlaces = 2           ' 显示两位小数  
        .Minimum = 1.3               ' 最小值  
        .Maximum = 9.7               ' 最大值  
        .Increment = 0.3              ' 每次单击的增量  
        .Value = 5.6                  ' 当前值  
    End With  
End Sub
```

运行上述程序，数值调节器中显示 5.60，当单击向上箭头，自动增加为 5.90，如图 5-16 所示。

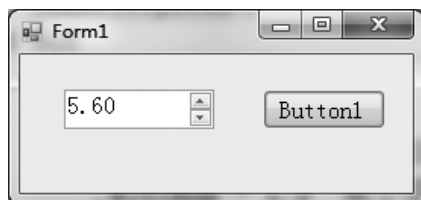


图 5-16 数值调节器

■ 5.5.4 DomainUpDown

DomainUpDown 控件的功能与 ComboBox 组合框控件是一样的，只是选择条目时不是下拉，而是像 NumericUpDown 一样单击上下箭头。

选择条目时触发 SelectedIndexChanged 事件。

```
Private Sub DomainUpDown()  
    With Me.DomainUpDown1  
        .Items.Clear()  
        .Items.AddRange({"Excel", "PowerPoint", "Word"})  
        .Text = "Outlook"  
        .Focus()  
    End With  
End Sub
```

启动窗体，单击按钮，自动向 DomainUpDown 控件中添加条目，如图 5-17 所示。

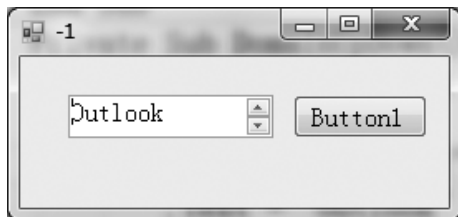


图 5-17 组合框形式的数值调节器

5.6 状态提示类控件

状态提示类控件有 NotifyIcon、ProgressBar、ToolTip 等。

5.6.1 NotifyIcon

NotifyIcon 控件用来在任务栏显示通知信息。

向窗体上添加 NotifyIcon 控件，在窗体上并不会看到该控件。

要在通知栏显示气球信息，必须为该控件设置 Icon，然后使用 ShowBalloonTipIcon 方法显示信息。

```
Private Sub NotifyIcon()  
    With Me.NotifyIcon1  
        .Icon = Icon.ExtractAssociatedIcon(filePath:="E:\Winsock\QQ.ico")  
        .ShowBalloonTip(timeout:=2, tipTitle:="友情提示", tipText:="试用已结束，  
请及时续费。", tipIcon:=ToolTipIcon.Info)  
    End With  
End Sub
```

运行上述程序，在屏幕右下角的通知栏弹出通知信息，如图 5-18 所示。

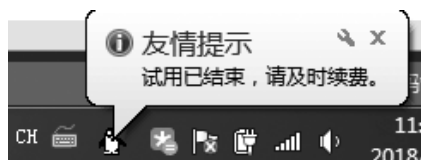


图 5-18 通知栏信息

5.6.2 ProgressBar

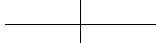
ProgressBar 是进度条控件，与 HScrollBar 类似，但是进度条控件的数值只能由程序代码改变，不能让用户拖动改变。

ProgressBar 控件的 Step 属性用来规定步进一次对进度条数值改变的程度，一般和 PerformStep 方法配合使用。

Increment 方法也是用来修改进度条数值的，不过该方法接收一个参数，用来指定修改的程度，该方法与 Step 属性的大小无关。

例如下面的程序，设置进度条的步长是 -5，那么执行一次 PerformStep 就会从现有数值基础上减去 5。

```
Private Sub ProgressBar()  
    With Me.ProgressBar1  
        .Minimum = 60  
        .Maximum = 100
```



```
.Step = -5  
.Value = 85  
End With  
End Sub
```

下面的程序，每单击一次按钮，进度条都会减少 5。

```
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click  
    With Me.ProgressBar1  
        .PerformStep()  
    End With  
End Sub
```

启动窗体，先单击“进度条设置”按钮，然后再单击“步进一次”按钮，会看到进度条的变化，如图 5-19 所示。

假设运行代码 `Me.ProgressBar1.Increment(15)`，进度条的数值会增加 15。



图 5-19 进度条控件

5.6.3 ToolTip

ToolTip 用来为指定控件提供提示语。向窗体上添加 ToopTip，该控件并不会出现在窗体上。

运行下面的程序，为窗体上的文本框 `TextBox1` 设置提示语。

```
Private Sub ToolTip()  
    With Me.ToolTip1  
        .ShowAlways = True  
        .ToolTipIcon = ToolTipIcon.Info  
        .ToolTipTitle = "重要提示"  
        .IsBalloon = False  
        .Show(text:="双击此文本框，可以快速输入当前时间。", window:=Me.TextBox1)  
    End With  
End Sub
```

启动窗体，单击“提示语设置”按钮，文本框附近出现提示信息，如图 5-20 所示。

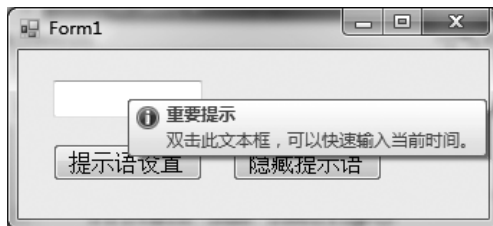


图 5-20 控件的提示信息

隐藏提示语的代码是：`Me.ToolTip1.Hide(win:=Me.TextBox1)`。



5.7 图片类控件

图片类控件有 PictureBox、ImageList 等。

5.7.1 PictureBox

PictureBox 是图片框控件，可以显示位图、图标、图元文件、增强型图元文件、JPEG 或 GIF 文件。

由于图片框控件自身的大小，与图片文件的大小一般情况下不一致，因此 PictureBox 的 SizeMode 属性用来设定显示方式。

当 SizeMode 设置为 AutoSize 时，图片文件的宽度和高度决定图片框控件的大小，图片很大，图片框自动调节自身大小以容纳图片。

当 SizeMode 设置为 StretchImage 时，图片框大小固定不变，而是将图片进行拉伸，使得图片能完全地显示在图片框控件中。

下面的程序，图片框的 SizeMode 为 AutoSize，自动装载计算机中的一个 bmp 图片。

```
Private Sub PictureBox()  
    With Me.PictureBox1  
        .BorderStyle = BorderStyle.FixedSingle  
        .SizeMode = PictureBoxSizeMode.AutoSize  
        '.SizeMode = PictureBoxSizeMode.StretchImage  
        .Image = Image.FromFile(filename:="E:\Chess\piece\bitmaps\horse.bmp")  
    End With  
End Sub
```

启动窗体，单击按钮，图片框控件中显示一个外部文件图片，如图 5-21 所示。

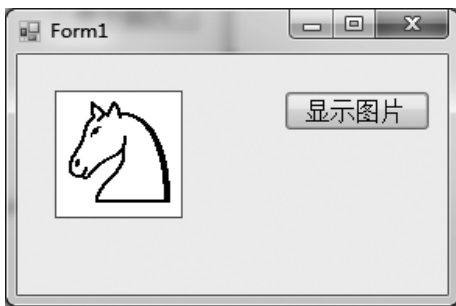


图 5-21 图片框控件

当外部文件被重命名或者删除以后，图片框不能正常显示图片。

5.7.2 ImageList

ImageList 可以看作是由多个图片构成的数组。可以把计算机中多个图片添加到该控件



中，以便于在程序的其他地方调用。

ImageList 控件添加到窗体后，并不会显示在窗体上。

ImageList 控件通常与 ListView、Treeview 等需要图标的其他控件配合使用，也可以为 PictureBox 控件、窗体等支持背景图片的对象提供图片。

ImageList 控件的重要属性是 ImageSize 类型数组，默认值是 16,16，如图 5-22 所示。

该属性值设置得越大，图片显示越大，但是最大值可以设置为 256,256。

下面的实例程序，在窗体上放置一个 ImageList 控件、两个 Button 控件。第一个按钮的作用是为 ImageList 控件添加若干图片，第二个按钮的作用是从 ImageList 控件中随机取出一张图片作为窗体的背景。

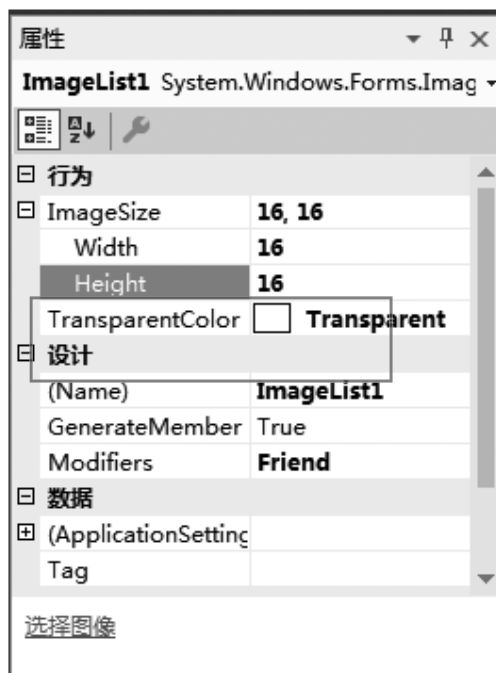


图 5-22 ImageList 控件的属性设置

```
Private Sub Button1_Click_1(sender As Object, e As EventArgs) Handles Button1.Click
    With Me.ImageList1
        .Images.Clear()
        .ImageSize = New Size(width:=128, height:=128)
        .Images.Add(Image.FromFile("E:\Picture\Fruit\苹果.jpg"))
        .Images.Add(Image.FromFile("E:\Picture\Fruit\香蕉.jpg"))
        .Images.Add(Image.FromFile("E:\Picture\Fruit\西瓜.jpg"))
    End With
End Sub

Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
    Randomize()
    Dim ImageIndex As Integer
    ImageIndex = Math.Round(Rnd() * (Me.ImageList1.Images.Count - 1), 0)
    With Me
        .BackgroundImageLayout = ImageLayout.Tile
        .BackgroundImage = .ImageList1.Images.Item(ImageIndex)
    End With
End Sub
```

代码分析：窗体的 BackgroundImageLayout 属性设置为 ImageLayout.Tile 是为了平铺图片，代码中的 ImageIndex 是由随机数经过取整得到的。

启动窗体，单击“装载图片”，然后单击“随机背景”，窗体的背景图片发生更换，如图 5-23 所示。



图 5-23 使用 ImageList 控件中的图片作为背景

5.8 日期时间类控件

日期时间类控件有 DateTimePicker、MonthCalendar、Timer 等。

5.8.1 DateTimePicker

DateTimePicker 控件提供一个日历，用户可以从该控件中选择日期和时间。

该控件的 Format 属性用来设置日历的格式，常用取值如下。

❑ Long、Short：显示日期。

❑ Time：显示时间。

❑ Custom：自定义格式，需要配合 CustomFormat 属性使用。

此外，该控件的 ShowUpDown 属性设置为 True，不显示日历，只在控件右侧显示上下增减箭头。

该控件的 Value 表示用户选择了的日期时间。

下面的程序，设置 DateTimePicker 控件的格式为日期、时间全部显示。

```
Private Sub DateTimePicker()  
    Dim SelectedDate As DateTime  
    With Me.DateTimePicker1  
        .Format = DateTimePickerFormat.Custom  
        .CustomFormat = "yyyy-MM-dd HH:mm:ss"  
        '.ShowCheckBox = True           ' 显示复选框  
        '.ShowUpDown = True           ' 显示增减箭头  
        SelectedDate = .Value  
    End With  
    MsgBox(SelectedDate.ToLongDateString)  
End Sub
```



启动窗体，单击按钮，然后从日历控件中选择日期、时间，如图 5-24 所示。

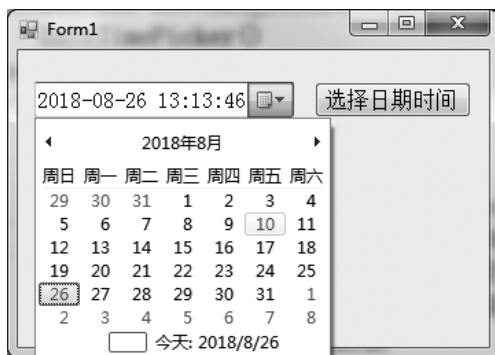


图 5-24 日历控件

5.8.2 Timer

Timer 是计时器控件，功能是隔一段时间就自动调用其 Tick 事件过程。

Timer 控件最重要的属性是 Enabled，该属性默认值是 False，当该属性为 True 时，自动循环不断地执行 Tick 事件过程，直至 Enabled 恢复为 False 为止。Interval 属性决定了执行 Tick 事件的间隔，单位是毫秒。

窗体上插入 Timer 控件，该控件并不会出现在窗体上。在使用该控件时，需要设置其 Enabled 属性和 Interval 属性，从而激活计时器。

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    With Me.Timer1
        .Interval = 1000          ' 隔一秒执行一次 Tick 事件
        .Enabled = True
    End With
    Me.ListBox1.Items.Clear()
End Sub
```

计时器被激活，会立即响应 Tick 事件过程。

```
Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
    Me.ListBox1.Items.Add(Now.ToLongTimeString)
End Sub
```

窗体上放置一个 Timer 控件、一个 ListBox 控件、一个 Button 控件。

窗体启动后，单击按钮，每隔一秒自动把当前时间添加到列表框中，如图 5-25 所示。

如果把 Timer 控件的 Enabled 属性设置为 False，就会自动停止运行 Tick 事件过程。

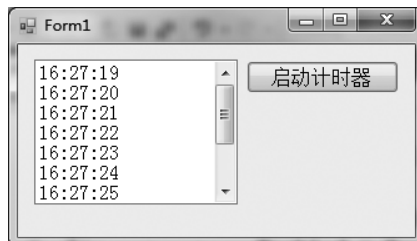


图 5-25 计时器控件

另外，使用 `Timer.Start` 方法，也可以自动设置 `Enabled` 为 `True`，并启动计时器事件。使用 `Timer.Stop` 方法，自动设置 `Enabled` 为 `False`，停止计时器。

■ 5.8.3 Stopwatch 对象

`Stopwatch` 不是一种控件，而是位于 `System.Diagnostics` 命名空间之下的一种对象，可以计算两个时间点经过的时间长度，起到计时器的作用。

`Stopwatch` 对象的主要方法如下。

- `Start`：开始或继续计时，不重置初始时间。
- `Stop`：停止计时。
- `Reset`：重置为零，并且停止计时。
- `Restart`：重置为零，并且开始计时。

`Stopwatch` 对象的主要属性如下。

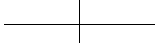
- `Elapsed`：计时器从开始到停止经历的时间，返回一个 `TimeSpan` 对象。
- `ElapsedMilliseconds`：计时器从开始到停止经历的时间，单位是毫秒。
- `IsRunning`：计时器是否在运行，返回布尔值。

下面的程序，演示了从 Winrar 压缩软件的官方网站下载文件到本地，使用计时器来测量下载用时。

项目实例 19 WindowsApp_Stopwatch 使用计时器

创建一个名为 `WindowsApp_Stopwatch` 的窗体应用程序，放置一个 `Button` 控件，窗体模块的代码如下。

```
Imports System.Diagnostics
Public Class Form1
    Private Watch1 As Stopwatch
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Watch1 = New Stopwatch()
        With Watch1
            .Start()
            Debug.WriteLine(" 运行状态: " & .IsRunning)
            My.Computer.Network.DownloadFile(address:="https://www.rarlab.com/rar/wrar561sc.exe", destinationFileName:="C:\temp\Winrar5.61SC.exe")
            Debug.WriteLine(" 下载完毕。")
            .Stop()
            Debug.WriteLine(" 运行状态: " & .IsRunning)
            Debug.WriteLine(" 总共用时 (毫秒): " & .ElapsedMilliseconds)
        End With
    End Sub
End Class
```



代码分析: My.Computer.NETWork.DownloadFile 可以方便地用于下载网页上的文件到本地, 在使用该方法前启动计时, 下载过程中窗体程序会进入阻塞状态, 下载完毕后自动停止计时器。

运行上述程序, 立即窗口的打印结果如下。

```
运行状态: True  
下载完毕。  
运行状态: False  
总共用时 (毫秒): 11834
```

5.9 其他控件

WebBrowser 控件用来显示网页内容, WindowsMediaPlayer 控件用来播放计算机中的音乐, PropertyGrid 控件用来显示和设置其他控件的属性。

5.9.1 WebBrowser

WebBrowser 控件用于在窗体中显示网页, 其 Navigate 方法接收一个网址的 url。

窗体上放置一个 TextBox、一个 Button、一个 WebBrowser 控件。

单击按钮, 浏览器控件自动打开文本框中指定的网址。

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click  
    With Me.WebBrowser1  
        .ScriptErrorsSuppressed = True ' 禁止弹出脚本错误对话框  
        .Navigate(Me.TextBox1.Text)  
    End With  
End Sub
```

启动窗体, 在文本框中输入网址, 单击“转到”按钮, 浏览器控件中显示网页, 如图 5-26 所示。



图 5-26 网页浏览器控件

WebBrowser 控件还有很多属性、方法和事件, 此处只给出最基本用法。

5.9.2 WindowsMediaPlayer

窗体上除了可以添加常用的控件以外，还可以使用其他高级的 ActiveX 控件，这些控件往往在控件工具箱中找不到。

本节的实例演示如何向窗体中添加 WindowsMediaPlayer 控件。

项目实例 20 WindowsApp_WindowsMediaPlayer 使用其他 ActiveX 控件

创建一个名为 WindowsApp_WindowsMediaPlayer 的窗体应用程序，在控件工具箱中右击，在右键菜单中选择“选择项”，如图 5-27 所示。

在弹出的“选择工具箱项”对话框中，切换至“COM 组件”选项卡，勾选 Windows Media Player，如图 5-28 所示。



图 5-27 查找更多控件

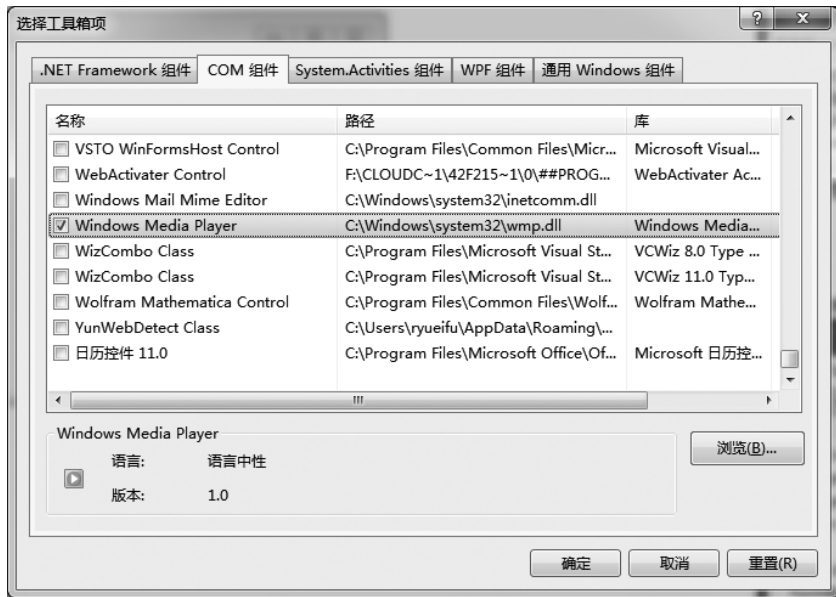


图 5-28 勾选 Windows Media Player 控件

单击对话框右下角的“确定”按钮，可以在控件工具箱下边找到该控件。

从控件工具箱把 Windows Media Player 控件拖放到窗体中，并且设置该控件的 Dock 属性为 Fill，如图 5-29 所示。

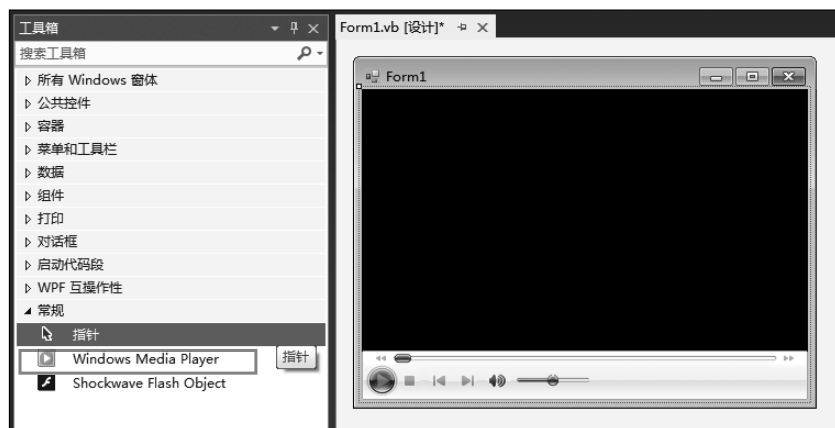
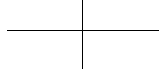


图 5-29 向窗体添加控件

然后设置窗体 Form1 的 Load 事件，实现启动窗体就自动播放计算机中的一个视频文件。

```
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles Me.Load
    With Me.AxWindowsMediaPlayer1
        .URL = "E:\行宝宝\最浪漫的事 - 苏妙玲 .mp4"
    End With
End Sub
```

启动窗体，出现一个播放视频的画面，如图 5-30 所示。

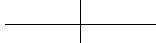


图 5-30 窗体中播放视频

5.9.3 PropertyGrid

一般情况下，窗体以及控件的属性，只能在窗体设计期间通过属性窗口进行修改设定，或者在窗体的运行期间通过程序代码访问属性。

使用 PropertyGrid 控件，可以实现把属性窗口嵌入到窗体中，即使在窗体运行期间仍然



可以手工修改属性窗口。

在控件工具箱中如果找不到 PropertyGrid 控件，可以在控件工具箱列表右击，在右键菜单中选择“选择项”，弹出“选择工具箱项”对话框。

切换到“.NET Framework 组件”选项卡，找到 PropertyGrid 并勾选，即可把该控件加入到控件工具箱中，如图 5-31 所示。

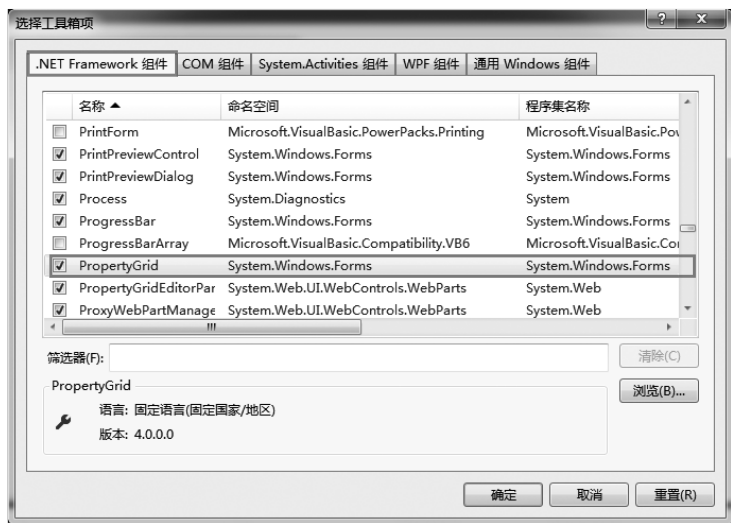


图 5-31 添加 PropertyGrid 控件到工具箱

PropertyGrid 控件可以为窗体上的其他控件提供属性设置，PropertyGrid 控件的 SelectedObject 属性需要指定一个控件。PropertySort 属性可以设置为按类别排序还是按字母排序。

窗体处于运行状态时，手工修改 PropertyGrid 中的属性值，就可以实时更改控件的属性，但是反过来如果控件的属性被修改，PropertyGrid 中相应的属性不会自动改变，需要调用 Refresh 方法刷新。

项目实例 21 WindowsApp_PropertyGrid 属性窗格控件

创建一个名为 WindowsApp_PropertyGrid 的窗体应用程序，窗体上放置一个 TextBox 控件、两个 Button 控件、一个 PropertyGrid 控件。

Button1 控件的功能是让 PropertyGrid 控件显示 TextBox1 的属性表。

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    With Me.PropertyGrid1
        .Dock = DockStyle.Right           ' 靠右停靠
        .SelectedObject = Me.TextBox1     ' 显示 TextBox1 的属性
        .PropertySort = PropertySort.Categorized ' 按类别排序
    End With
End Sub
```



Button2 控件的功能是让 PropertyGrid 属性表同步 TextBox1 的属性。

```
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
    Me.PropertyGrid1.Refresh()
End Sub
```

启动窗体，单击“初始设置”按钮，属性表中显示文本框的属性。在文本框中手工输入一些内容，然后单击“刷新”按钮，属性表中自动更新，如图 5-32 所示。

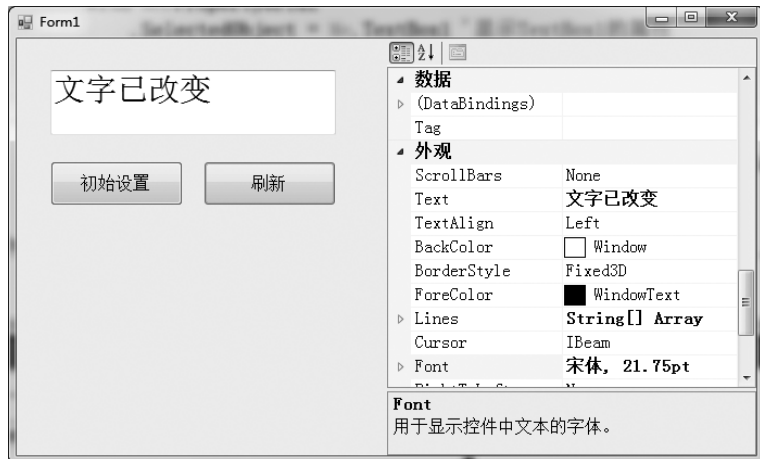


图 5-32 运行期间的属性窗口

5.9.4 FileSystemWatcher

FileSystemWatcher 是一个用于实时监控文件系统变化的控件。

为该控件设置好监视路径，然后启动监视。被监视的文件夹中的文件或路径发生了改变，就会通知窗体应用程序。

向窗体中添加 FileSystemWatcher 控件的方法是，在控件工具箱中展开“组件”，就可以找到该控件，然后拖放到窗体中即可，如图 5-33 所示。

下面的实例程序，演示了启动监视后，文件一发生改变就向窗体上的 RichTextbox 控件中添加日志。

项目实例 22 WindowsApp_FileSystemWatcher 文件监视

创建一个名为 WindowsApp_FileSystemWatcher 的窗体应用程序，窗体上放置一个 FileSystemWatcher 控件、一个 TextBox 控件、一个 Button 控件、一个 RichTextbox 控件。

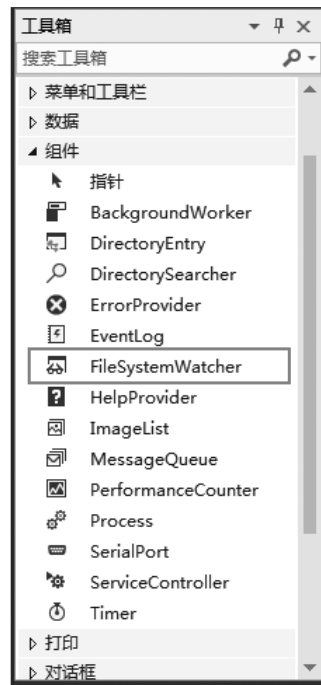


图 5-33 FileSystemWatcher 控件