

VisualBasic工具栏、对话框 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/137/2021_2022_VisualBasi_c97_137556.htm 工具栏 可以用工具栏进一步增强应用程序的菜单界面。工具栏含有工具栏按钮，它提供了对于应用程序中最常用的命令的快速访问。例如：Visual Basic 的工具栏含有执行常用命令的 ToolBarButton，如“打开”（打开已存在的工程）、“保存”（存储当前的工程），等等。创建工具栏工具栏（也称为发条或者控制栏）已经成为许多基于 Windows 的应用程序的标准功能。工具栏提供了对于应用程序中最常用的菜单命令的快速访问。使用 ToolBar 控件来创建工具栏非常容易且很方便，它在 Visual Basic 的专业版与企业版中是可用的。如果使用的是 Visual Basic 学习版，则可以像本章后面的“协调菜单与工具栏的外观”所描述的那样用手工来创建工具栏。下列示例演示为 MDI 应用程序创建工具栏的过程；在标准窗体上创建一个工具栏的过程基本上一样。要手工创建工具栏，请按照以下步骤执行：1. 在 MDI 窗体上放置一个图片框。图片框的宽度会自动伸展，直到填满 MDI 窗体工作空间。工作空间就是窗体边框以内的区域，不包括标题条、菜单栏或所有的工具栏、状态栏或者可能在窗体上的滚动条。注意 只能把那些直接支持 Align 属性的控件放置在 MDI 窗体上（图片框是支持这一属性的唯一的标准控件）。2. 在图片框中，可以放置任何想在工具栏上显示的控件。典型地，用 CommandButton 或 Image 控件来创建工具栏按钮。图 6.16 表示出了一个含有 Image 控件的工具栏。要在图片框中添加控件，单击工具栏中的控件按钮，然后在图片框中

画出它。注意当 MDI 窗体中包含了图片框时，该 MDI 窗体的内部区域不包括图片框在内。例如：MDI 窗体的 ScaleHeight 属性返回 MDI 窗体的内部高度，这个高度已不包括图片框的高度。

3. 设置设计时属性。使用工具栏的一个好处是可以显示一个形象的命令图示。Image 控件是作为工具栏按钮的一个很好的选择，因为可以用它来显示一个位图。在设计时设置其 Picture 属性来显示一个位图；这样，当该按钮被单击时，即能提供一个命令执行的可见信息。也可以通过设置按钮的 ToolTipText 属性来使用工具提示，这样，当用户把鼠标指针保持在一个按钮上时，就可以显示出该工具栏按钮的名称。

4. 编写代码 因为工具栏频繁地用于提供对其它命令的快捷访问，因而在大部分时间内都是从每一个按钮的 Click 事件中调用其它过程，比如对应的菜单命令。提示可用一个不显示工具栏的 MDI 窗体来使用在运行时不可见的控件（比如 Timer 控件）。为了做到这一点，在 MDI 窗体上放置一个图片框，把控件放到图片框中，然后把图片框的 Visible 属性设置为 False。

编写工具栏的代码 工具栏用于提供访问某些应用程序命令的快捷方法。例如：图 6.16 中工具栏上的第一个按钮就是“新建文件”命令的快捷键。现在，可以在 MDI NotePad 示例应用程序中三个地方请求创建新文件。

1. 在 MDI 窗体上（MDI 窗体上“文件”菜单中的“新建”命令）。
2. 在子窗体上（在子窗体“文件”菜单上的“新建”命令）。
3. 在工具栏上（“新建文件”按钮）。

与其把这个代码重复三次，还不如从子窗体的 mnuFileNew_Click 事件取出原代码，然后把它放入子窗体的一个公用过程中。可以从上面任何一个事件过程调用这个过程。以下是一个示例：

这个例程在公共过程中。 Public Sub FileNew () Dim frmNewPad As New frmNotePad frmNewPad.Show End Sub 在子窗体的“文件”菜单上选取“新建”。 Private Sub mnuchildFileNew_Click () FileNew End Sub 在 MDI 窗体的“文件”菜单上选取“新建”。 Private Sub mnumdiFileNew_Click () frmNotePad.FileNew End Sub 在工具栏上单击“新建文件”按钮。 Private Sub btnFileNew_Click () frmNotePad.FileNew End Sub 协调菜单与工具栏的外观 当由其它应用程序提供的对象在窗体中被激活时，有许多方法让对象的菜单与工具栏出现在容器窗体内；然而，需要规定它们将如何显示。这一过程叫做用户界面协调，因为 Visual Basic 与已经链接或嵌入的对象必须协调容器窗体中的空间。 控制菜单的外观 通过设置窗体的 NegotiateMenus 属性可以决定一个链接或嵌入的对象的菜单是否出现在容器窗体中。如果子窗体的 NegotiateMenus 属性置为 True（缺省），而且容器有一个定义的菜单栏，那么，当这个对象被激活时，它的菜单就放置在容器的菜单栏中。如果容器没有菜单栏，或者 NegotiateMenus 属性被设置为 False，那么，当这个对象激活时对象的菜单将不出现。注意 NegotiateMenus 属性不适用于 MDI 窗体。 控制工具栏的外观 MDI 窗体的 NegotiateToolbars 属性决定了链接或嵌入对象的工具栏是不固定的调色板还是被放置在父窗体上。这种性能不要求工具栏出现在 MDI 父窗体上。如果 MDI 窗体的 NegotiateToolbars 属性设为 True，则对象的工具栏出现在 MDI 父窗体上。如果 NegotiateToolbars 设为 False，则对象的工具栏就为不固定的调色板。注意 NegotiateToolbars 属性只用于 MDI 窗体。如果 MDI 窗体包含工具栏，它通常被包含

在父窗体的 PictureBox 控件中。图片框的 Negotiate 属性决定了被激活时容器的工具栏是继续显示还是被对象的工具栏所代替。如果 Negotiate 设置为 True，则除了容器的工具栏外还显示对象的工具栏。如果 Negotiate 设置为 False，则对象的工具栏代替容器的工具栏。注意菜单与工具栏的协调只发生在支持现场激活的可插入对象之上。有关现场激活的详细信息，请参阅第十章“用部件编程”。用以下过程可以看到这三个属性如何交互作用。要执行菜单与工具栏的协调，请按照以下步骤执行：1. 在 MDI 窗体中添加工具栏。这在本章前面的“创建工具栏”中描述过。2. 在子窗体上放置一个可插入的对象。3. 设置 NegotiateMenus、NegotiateToolbars 以及 Negotiate 属性。4. 运行此应用程序，然后双击该对象。

对话框在基于 Windows 的应用程序中，对话框被用来：提示用户提供应用程序继续执行所需要的数据。向用户显示信息。例如，在 Visual Basic 中，用“打开文件”对话框来显示已存在的工程。Visual Basic 中的“关于”对话框也是一个如何使用对话框来显示信息的例子。在菜单栏上单击“帮助”，选择“关于 Visual Basic”菜单项，则显示“关于”对话框。模式与无模式的对话框 对话框不是模式就是无模式的。模式对话框，在可以继续操作应用程序的其他部分之前，必须被关闭（隐藏或卸载）。例如，如果一个对话框，在可以切换到其它窗体或对话框之前要求先单击“确定”或“取消”，则它就是模式的。Visual Basic 中的“关于”对话框是模式的。显示重要消息的对话框总应当是模式的那就是说，在继续做下去之前，总是要求用户应当先关上对话框或者对它的消息作出响应。无模式的对话框允许在对话框与其他窗体之间转移焦

点而不用关闭对话框。当对话框正在显示时，可以在当前应用程序的其他地方继续工作。无模式对话框很少使用。Visual Basic 中“编辑”菜单中的“查找”对话框就是一个无模式对话框的实例。无模式对话框用于显示频繁使用的命令与信息。要将窗体作为模式对话框显示，请 1.使用 Show 方法，其 style 参数值为 vbModal（一个值为 1 的常数）。例如：将 frmAbout 作为模式对话框显示。frmAbout.Show vbModal 要将窗体作为无模式对话框显示，请 2.使用不带 style 参数的 Show 方法。例如：将 frmAbout 作为无模式对话框显示。

frmAbout.Show 注意 如果窗体显示为模式对话框，则只有当对话框关闭之后，在 Show 方法后的代码才能执行。然而，当窗体被显示为无模式对话框时，在该窗体显示出来以后，Show 方法后面的代码紧接着就会执行。Show 方法另有一个可选参数，owner，可用来指定窗体的父子关系。可将某个窗体名传给这个参数，使得这个窗体成为新窗体的拥有者。要显示一个窗体，作为另一个窗体的子窗体，请 这里 Show 方法有两个参数：style 和 owner。例如：将 frmAbout 显示为 frmMain 的无模式子窗体。frmAbout.Show vbModeless, frmMain 在 Show 方法中使用 owner 参数，确保对话框在它的父窗体最小化时它也最小化，或者在其父窗体关闭时它也卸载。

预定义对话框的使用 在应用程序中添加对话框最容易的方法是使用预定义对话框，因为不必考虑设计、装载或者显示对话框方面的问题。然而，控件在其外观上要受到限制。预定义的对话框总是模式的。下表列出了 Visual Basic 应用程序中添加预定义对话框时能使用的函数。用输入框来提示输入应用 InputBox 函数请求提供数据。这个函数显示要求输入

数据的模式对话框。图 6.17 所示的文本输入框提示输入要打开文件的名称。以下代码显示图 6.17 中所示的输入框。

```
FileName = InputBox("Enter file to open:", "File Open")
```

注意 切记当使用 InputBox 函数时，对对话框的各部分的控制非常有限。只能改变标题栏中的文本、显示给用户的命令提示、对话框在屏幕上的位置以及它是否显示一个“帮助”按钮。详细信息请参阅《语言参考》的“InputBox 函数”。用消息对话框显示信息可以用 MsgBox 函数获得“是”或者“否”的响应，并显示简短的消息，比如：错误、警告或者对话框中的期待。看完这些消息以后，可选取一个按钮来关闭该对话框。如果文件不能打开，那么名为 Text Editor 的应用程序就会显示如图 6.18 所示的消息对话框。以下代码显示如图 6.18 所示的消息框：

```
MsgBox "Error encountered while trying to open file, _ please retry.", vbExclamation, "Text Editor"
```

注意 所谓模式的，既可以局限于应用程序中，也可以局限于系统中。如果消息框的模式局限在应用程序中（缺省），则在这个对话框未消失之前不能切换到该应用程序的其他部分，但是可以切换到其他应用程序。在消息框未消失之前系统的模式消息框不允许切换到别的应用程序。详细信息请参阅《语言参考》的“MsgBox 函数”。

用窗体作为自定义对话框 自定义对话框就是用户所创建的含有控件的窗体这些控件包括命令按钮、选取按钮和文本框它们可以为应用程序接收信息。通过设置属性值来自定义窗体的外观。也可以编写在运行时显示对话框的代码。要创建自定义对话框，可以从新窗体着手，或者自定义现成的对话框。如果重复过多，可以建造能在许多应用程序中使用的对话框的集合。要自定义现存的对话框，请

按照以下步骤执行：1. 从“工程”菜单中选取“添加窗体”，在工程中添加一现存的窗体。2. 从“文件”菜单中选取“filename 另存为”并输入新的文件名（这可以防止改变已存在的窗体版本）。3. 根据需要自定义窗体的外观。4. 在代码窗口中自定义事件过程。要创建新的对话框，请按照以下步骤执行：1. 从“工程”菜单中选取“添加窗体”。- 或者 - 在工具栏上单击“窗体”按钮，创建新的窗体。2. 如有必要，自定义窗体外观。3. 在“代码”窗口中自定义事件过程。有很大的自由来定义自定义对话框的外观。它可以是固定的或可移动的、模式或无模式的。它可以包含不同类型的控件；然而，对话框通常不包括菜单栏、窗口滚动条、最小化与最大化按钮、状态条或者尺寸可变的边框。本主题余下的部分将讨论创建典型类型的对话框方法。添加标题对话框总应当有标识它的标题。要创建标题，设置该窗体的 Caption 属性为将在标题条中显示的文本字符串。通常，这一步是在设计时使用“属性”窗口来完成的，但也可以用代码来完成这一步。例如：`frmAbout.Caption = "About"` 提示 如果想完全删除此标题栏，可以设置窗体的 `ControlBox`、`MinButton` 和 `MaxButton` 为 `False`；设置 `BorderStyle` 为尺寸不可变的设置（0、1 或 3），并设置 `Caption` 为空字符串（""）。设置标准对话框的属性一般来说，用户响应对话框时，先提供信息，然后用“确定”或者“取消”命令按钮关闭对话框。因为对话框是临时性的，用户通常不需要对它进行移动、改变尺寸、最大化或最小化等操作。其结果是：随新窗体出现的可变尺寸边框类型、“控制”菜单框、“最大化”按钮以及“最小化”按钮，在大多数对话框中都是不需要的。通过设置

BorderStyle、ControlBox、MaxButton 和 MinButton 属性，可以删除这些项目。例如，“关于”对话框可能使用以下的属性设置。记住，如果删除“控制”菜单框 (ControlBox = False)，则必须向用户提供退出该对话框的其它方法。实现的办法通常是在对话框中添加“确定”、“取消”或者“退出”命令按钮，并在隐藏或卸载该对话框的 Click 按钮事件中添加代码。添加和放置命令按钮 模式对话框必须至少包含一个退出该对话框的命令按钮。通常用两个命令按钮：其中一个按钮开始动作，而另一个按钮关闭该对话框而不做任何改变。典型状态是，这两个按钮的 Caption 属性设置“确定”与“取消”。在这个方案中，“确定”命令按钮的 Default 属性设置为 True，而 Cancel 命令按钮的 Cancel 属性设置为 True。虽然“确定”与“取消”是最常用的按钮，其他的按钮标题组合也可使用。显示消息的对话框通常使用 Label 控件来显示错误消息或者命令提示，并且用一至两个命令按钮来执行动作。例如，也许给标识的 Caption 属性赋以错误消息或者命令提示，而给两个命令按钮控件的 Caption 属性指定“是”与“否”。当用户选取“是”，则发生一个动作，当选取“否”时，则发生另一个动作。这种类型对话框的命令按钮通常被放置在对话框的底部或右边，而顶部或左边的按钮为缺省按钮，如图 6.19 所示。设置缺省、取消和焦点 命令按钮控件提供了以下的属性：Default Cancel TabIndex TabStopDefault 按钮就是当用户按下 ENTER 键时选中的按钮。在一个窗体上，只能有一个命令按钮的 Default 属性可以设置为 True。按下 ENTER 键就调用缺省命令按钮的 Click 事件。这个功能连同编辑控件（比如 TextBox）一起工作。例如，可以在文本框中键入数

据，然后按下 ENTER 键来产生 Click 事件，以取代选取“确定”按钮。“取消”按钮就是当按下 ESC 键时选中的按钮。在一个窗体上，只能有一个命令按钮的 Cancel 属性可以设置为 True。按下 ESC 键调用“取消”命令按钮的 Click 事件。

“取消”按钮也可以为缺省命令按钮。要指定对话框的“取消”按钮，设置该命令按钮的 Cancel 属性为 True。提示一般说来，代表最可靠的或者最安全的操作的按钮应当是缺省按钮。例如，在“文本替换”对话框中，“取消”应当是缺省按钮，而不是“全部替换”。也可以指定当对话被显示时具有焦点的按钮。当窗体被显示时，带有最低的 TabIndex 设置的控件接收焦点。按下 ENTER 键会调用缺省命令按钮或者有焦点的命令按钮的 Click 事件。要在窗体被显示时将焦点给某个命令按钮，则设置该命令按钮的 TabIndex 为 0，它的 TabStop 属性为 True。也可以用 SetFocus 方法在窗体被显示时将焦点给指定的控件。详细信息请参阅《语言参考》的“TabIndex 属性”与“TabStop 属性”。

使对话框上的控件无效 有时候需要使控件无效，因为它们的动作在当前的上下文中不适用。例如，当 Visual Basic 的“查找”对话框第一次显示时，“查找下一个”按钮应无效，如图 6.20 所示。设置对话上的控件的 Enabled 属性为 False，可以使其无效。要使对话框上的控件无效，请 1. 设置每个控件的 Enabled 属性为 False。例如：`cmdFindNext.Enabled = False` `cmdReplace.Enabled = False`

显示自定义对话框 使用应用程序中显示其他窗体同样的方法来显示对话框。当应用程序运行时，启动窗体会自动装入。想在应用程序中出现第二个窗体或对话框，要用装入并显示它的代码来实现。同样，想要窗体或对话框消失，也

要编写卸载或隐藏它的代码。以下代码在用户选取“帮助”菜单中的“关于”菜单项时显示“关于”对话框。Private Sub mnuHelpAbout_Click () 此处用 style = vbModal 的 Show 方法将对话框显示为模式的。frmAbout.Show vbModal End Sub

100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com