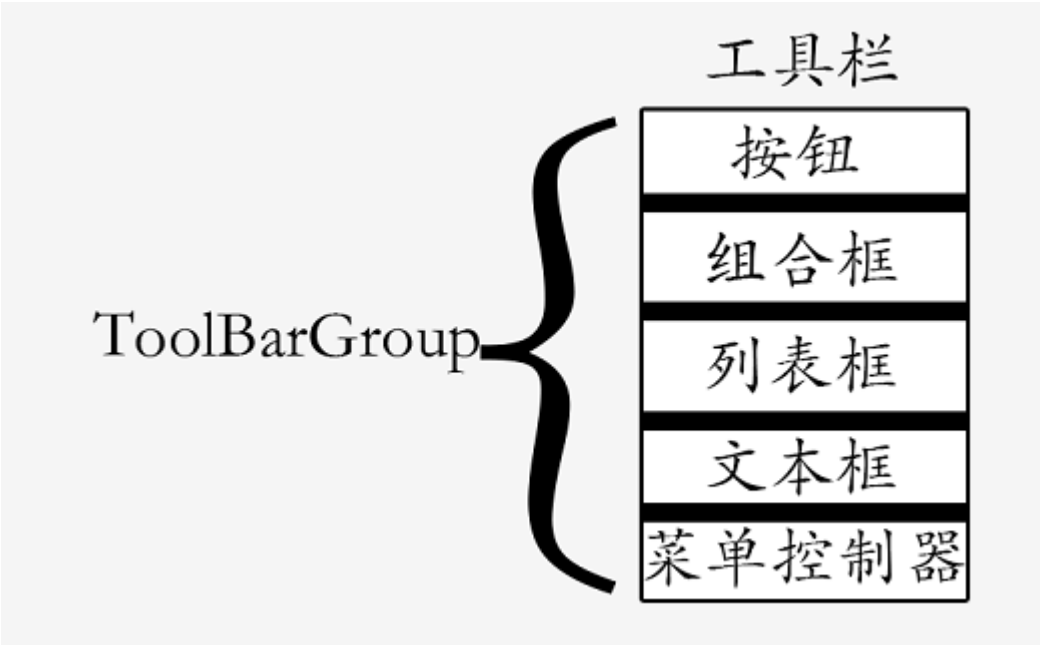


扩展工具栏

工具栏是一个包含绑定命令的按钮的水平或垂直条。具体取决于其实现，在 IDE 中的工具栏可以重新定位、也可以停靠在主 IDE 窗口的任何一侧或保持在其他窗口前面。此外，用户可以将命令添加到工具栏。通常情况下，在 Vspackage 中的工具栏是用户可自定义的。



如上图，工具栏通常是按钮和其他控件（如组合框、列表框、文本框和菜单控制器）组成的行。在工具栏的按钮都有图标，这样会更形象化。那么怎么把这些控件关联起来呢？我们可以把这些控件放到一个工具栏组中，然后让这个组的Parent为工具栏，这样就在这个组和工具栏关联起来了，这些控件元素也就添加到工具栏中了。

当我们把自定义的控件添加到工具栏后，怎么使它触发相应的操作，实现相应的功能？

所有的工具栏的控件都与命令相关联，比如工具栏的按钮，当我们单击工具栏的按钮时，就会激活其相关命令。工具栏按钮通常是具有提示基础命令的图标，如用于打印命令的打印机。在下拉列表控件中，列表中的每一项都与不同的命令关联。

为 IDE 创建工具栏

创建一个名为的 VSIX 项目 `IDEToolbar`。添加一个名为的菜单命令项模板 `ToolbarTestCommand`。

1.首先定义一个工具栏和工具栏组，在 `ToolbarTestCommandPackage.vsct`，查找符号部分。在名为 `guidToolbarTestCommandPackageCmdSet GuidSymbol` 元素中，添加一个工具栏和工具栏组的声明，如下所示。

```
<IDSymbol name="Toolbar" value="0x1000" />
<IDSymbol name="ToolbarGroup" value="0x1050" />
```

2.在命令部分顶部创建菜单部分。将菜单元素添加到要定义您的工具栏上的菜单部分。这里不需要分配父组。此外，也不需要需设置优先级，因为用户可以移动工具栏。

```

<Menus>
  <Menu guid="guidToolbarTestCommandPackageCmdSet" id="Toolbar"
    type="Toolbar" >
    <CommandFlag>DefaultDocked</CommandFlag>
    <Strings>
      <ButtonText>Test Toolbar</ButtonText>
      <CommandName>Test Toolbar</CommandName>
    </Strings>
  </Menu>
</Menus>

```

3.添加一个工具栏组，这个组用来存放其他控件，这个组的Parent设置为Toolbar,就把组和工具栏关联起来，从而实现将其他控件元素添加到工具栏

```

<Group guid="guidToolbarTestCommandPackageCmdSet" id="ToolbarGroup" priority="0x0000">
  <Parent guid="guidToolbarTestCommandPackageCmdSet" id="Toolbar"/>
</Group>

```

4.接下来定义在工具栏上要显示的按钮。把按钮的Parent设置为ToolbarGroup,这样就把这个按钮控件添加到这个组中了

```

<Button guid="guidToolbarTestCommandPackageCmdSet" id="ToolbarTestCommandId" priority="0x0100"
type="Button">
  <Parent guid="guidToolbarTestCommandPackageCmdSet" id="ToolbarGroup" />
  <Icon guid="guidImages" id="bmpPic1" />
  <Strings>
    <ButtonText>Invoke ToolbarTestCommand</ButtonText>
  </Strings>
</Button>

```

默认情况下，如果工具栏不具有任何命令，则不会不显示。

生成项目并启动调试。应显示的实验实例。

右键单击 Visual Studio 菜单栏，可以获取工具栏的列表。选择**Test Toolbar**。

工具栏上出现了相应的图标。当您单击该图标时，您应看到一个消息框，指示ToolbarTestCommandPackage。内部 IDEToolbar.ToolbarTestCommand.MenuItemCallback()。

创建工具窗口

Visual Studio 2010 和更高版本中的所有工具窗口都是基于 WPF 的。工具窗口不同于文档窗口，文档窗口在读写模式下显示文件。工具窗口是Visual Studio IDE的子窗口。和“工具箱”、“解决方案资源管理器”、“属性”窗口和“Web 浏览器”一样，它可以停靠、浮动或者以标签的形式显示在文档框架中。工具窗口框架由 IDE 提供，用于控制大小、位置、停靠状态和其他持久性属性。工具窗口窗格用于显示内容。工具窗口窗格可以承载 WPF 用户控件，并支持工具栏。你可以重写 Window属性以返回所承载的控件的句柄。

工具窗口默认情况下是单实例，这意味着一次只能打开一个工具窗口的实例。打开单实例工具窗口后，它保持打开状态直到关闭 IDE。当你单击单实例工具窗口上的关闭按钮时，仅更改其可见性。你还可以创建多实例工具窗口，以便可以同时打开窗口中的多个实例。

1. 创建一个名为**FirstWindow**的 VSIX 项目。

2. 当项目打开后时，添加一个名为的工具窗口项模板**MyWindow**。在**解决方案资源管理器**，用鼠标右键单击项目节点并选择**添加 / 新项**。在添加新项对话框中，转到**Visual C# / 可扩展性**，然后选择**自定义工具窗口**。在名称在窗口的底部字段中，工具窗口文件名称更改为**MyWindow.cs**。

创建好项目之后，在项目中MyWindowControl.xaml中会默认有个TextBlock，它的值为MyWindow，还有一个button，button里面写着Click me! button还有个Click事件如下：

```
private void button1_Click(object sender, RoutedEventArgs e)
{
    MessageBox.Show(string.Format(System.Globalization.CultureInfo.CurrentUICulture, "Invoked '{0}'", this.ToString()), "MyWindow");
}
```

在.vsct文件中，默认定义了一个命令如下：

```
<GuidSymbol name="guidMyWindowPackageCmdSet" value="{318f4fb8-f12f-4a05-b93b-c9b7fccc07a6}">
    <IDSymbol name="MyWindowCommandId" value="0x0100" />
</GuidSymbol>
```

在Commands中默认的button定义如下：

```
<Button guid="guidMyWindowPackageCmdSet" id="MyWindowCommandId" priority="0x0100" type="Button">
    <Parent guid="guidSHLMainMenu" id="IDG_VS_WNDO_OTRWNDWS1"/>
    <Icon guid="guidImages" id="bmpPic1" />
    <Strings>
        <ButtonText>MyWindow</ButtonText>
    </Strings>
</Button>
```

3. 生成项目并启动调试。

遇到了下面的问题：

“VSCTCompiler”任务意外失败。System.OutOfMemoryException: 内存不足。

解决方法：模版提供的默认ico和png有问题，不能用，替换了就可以运行了

继续生成项目并启动调试，将显示 Visual Studio 的实验实例。

4. 在实验实例中，转到视图 / 其他窗口。

可以看到的菜单项**MyWindow**。单击该按钮。

可以看到一个工具窗口具有标题**MyWindow**和按钮的一种说法Click Me !

