向工具栏添加菜单控制器[[1]](#footnote-0)

原文：

<https://docs.microsoft.com/zh-cn/visualstudio/extensibility/adding-a-menu-controller-to-a-toolbar>

本文内容

准备

创建菜单控制器

实现菜单控制器命令

测试菜单控制器

另见

本指导以[向工具窗口添加工具栏](https://docs.microsoft.com/zh-cn/visualstudio/extensibility/adding-a-toolbar-to-a-tool-window)为基础，演示如何向工具窗口工具栏添加菜单控制器。这里将要演示的步骤也可应用于[添加一个工具栏](https://docs.microsoft.com/zh-cn/visualstudio/extensibility/adding-a-toolbar)里创建的工具栏。

菜单控制器是一个可拆分控件。它的左边部分显示了最近使用过的命令，可以通过点击它来运行。右侧有一个箭头，点击可显示附加的命令的列表。单击列表上的命令就会执行命令，并替换菜单控制器左边的命令。这样，菜单控制器的操作就像一个命令按钮，只是它总是显示列表中最后一个使用过的命令。

可以把菜单控制器放在菜单上，但它们最常放在工具栏上。

准备

启动Visual Studio 2015,请不要在下载中心安装Visual Studio SDK。Visual Studio SDK是Visual Studio的可选功能。您可以稍后安装VS SDK。详见[安装Visual Studio SDK](https://docs.microsoft.com/zh-cn/visualstudio/extensibility/installing-the-visual-studio-sdk)。

创建菜单控制器

创建菜单控制器

**1.** 按照[向工具窗口添加工具栏](https://docs.microsoft.com/zh-cn/visualstudio/extensibility/adding-a-toolbar-to-a-tool-window)的步骤创建一个带有工具栏的工具窗口。

**2.** 在TWTestCommandPackage.vsct里找到Symbols部分。在名为**guidTWTestCommandPackageCmdSet**的GuidSymbol元素里声明菜单控制器、菜单控制器组和三个菜单项。

<IDSymbol name="TestMenuController" value="0x1300" />

<IDSymbol name="TestMenuControllerGroup" value="0x1060" />

<IDSymbol name="cmdidMCItem1" value="0x0130" />

<IDSymbol name="cmdidMCItem2" value="0x0131" />

<IDSymbol name="cmdidMCItem3" value="0x0132" />

**3.** 在Menus节点里最后一项菜单后定义菜单控制器，就把它定义为一个菜单。

<Menu guid="guidTWTestCommandPackageCmdSet" id="TestMenuController" priority="0x0100"

type="MenuController">

<Parent guid="guidTWTestCommandPackageCmdSet" id="TWToolbarGroup" />

<CommandFlag>IconAndText</CommandFlag>

<CommandFlag>TextChanges</CommandFlag>

<CommandFlag>TextIsAnchorCommand</CommandFlag>

<Strings>

<ButtonText>Test Menu Controller</ButtonText>

<CommandName>Test Menu Controller</CommandName>

</Strings>

</Menu>

菜单控制器必须包含TextChanges和TextIsAnchorCommand标志，以便显示上次选择的命令。

**4.** 在Groups节点里最后一项组后添加菜单控制器组。

<Group guid="guidTWTestCommandPackageCmdSet" id="TestMenuControllerGroup"

priority="0x000">

<Parent guid="guidTWTestCommandPackageCmdSet" id="TestMenuController" />

</Group>

只要将菜单控制器设置为父对象，此组中的任何命令都会出现在菜单控制器中。省略priority属性会被默认设置为0，因为它是菜单控制器中唯一的组。

**5.** 在Buttons节点最后一个按钮后面为每个菜单项添加一个Button元素。

<Button guid="guidTWTestCommandPackageCmdSet" id="cmdidMCItem1" priority="0x0000"

type="Button">

<Parent guid="guidTWTestCommandPackageCmdSet" id="TestMenuControllerGroup" />

<Icon guid="guidImages" id="bmpPic1" />

<CommandFlag>IconAndText</CommandFlag>

<Strings>

<ButtonText>MC Item 1</ButtonText>

<CommandName>MC Item 1</CommandName>

</Strings>

</Button>

<Button guid="guidTWTestCommandPackageCmdSet" id="cmdidMCItem2" priority="0x0100"

type="Button">

<Parent guid="guidTWTestCommandPackageCmdSet" id="TestMenuControllerGroup" />

<Icon guid="guidImages" id="bmpPic2" />

<CommandFlag>IconAndText</CommandFlag>

<Strings>

<ButtonText>MC Item 2</ButtonText>

<CommandName>MC Item 2</CommandName>

</Strings>

</Button>

<Button guid="guidTWTestCommandPackageCmdSet" id="cmdidMCItem3" priority="0x0200"

type="Button">

<Parent guid="guidTWTestCommandPackageCmdSet" id="TestMenuControllerGroup" />

<Icon guid="guidImages" id="bmpPicSearch" />

<CommandFlag>IconAndText</CommandFlag>

<Strings>

<ButtonText>MC Item 3</ButtonText>

<CommandName>MC Item 3</CommandName>

</Strings>

</Button>

**6.** 到此为止就可以看到菜单控制器了。生成解决方案然后开始调试。会打开实验实例。

a. 在**视图 / 其他窗口**菜单里打开**Test ToolWindow**。

b. 菜单控制器就会出现在工具窗口的工具栏里。

c. 点击菜单控制器右侧箭头，可以看到三个命令。

注意点击命令后，菜单控制器的标题就会显示这个命令。在下一节中，我们将添加激活这些命令的代码。

实现菜单控制器命令

**1.** 在TWTestCommandPackageGuids.cs[[2]](#footnote-1)里现有命令ID后为三个菜单项添加命令ID。

public const int cmdidMCItem1 = 0x130;

public const int cmdidMCItem2 = 0x131;

public const int cmdidMCItem3 = 0x132;

**2.** 在TWTestCommand.cs里TWTestCommand类顶部添加如下代码。

private int currentMCCommand; // The currently selected menu controller command

**3.** 在TWTestCommand构造器里的最后一次调用AddCommand方法后，用相同的处理程序为每个命令添加路由事件的代码。

for (int i = TWTestCommandPackageGuids.cmdidMCItem1;

i <= TWTestCommandPackageGuids.cmdidMCItem3; i++)

{

CommandID cmdID = new

CommandID(new Guid(TWTestCommandPackageGuids.guidTWTestCommandPackageCmdSet), i);

OleMenuCommand mc = new OleMenuCommand(new EventHandler(OnMCItemClicked[[3]](#footnote-2)), cmdID);

mc.BeforeQueryStatus += new EventHandler(OnMCItemQueryStatus);

commandService.AddCommand(mc);

// The first item is, by default, checked.

if (TWTestCommandPackageGuids.cmdidMCItem1 == i)

{

mc.Checked = true;

this.currentMCCommand = i;

}

}

**4.** 为TWTestCommand类添加事件处理程序来把选过命令标记为选中（Checked）。

private void OnMCItemQueryStatus(object sender, EventArgs e)

{

OleMenuCommand mc = sender as OleMenuCommand;

if (null != mc)

{

mc.Checked = (mc.CommandID.ID == this.currentMCCommand);

}

}

**5.** 添加用户选择菜单控制器里命令的事件处理程序来显示一个对话框。

private void OnMCItemClicked(object sender, EventArgs e)

{

OleMenuCommand mc = sender as OleMenuCommand;

if (null != mc)

{

string selection;

switch (mc.CommandID.ID)

{

case TWTestCommandPackageGuids.cmdidMCItem1:

selection = "Menu controller Item 1";

break;

case TWTestCommandPackageGuids.cmdidMCItem2:

selection = "Menu controller Item 2";

break;

case TWTestCommandPackageGuids.cmdidMCItem3:

selection = "Menu controller Item 3";

break;

default:

selection = "Unknown command";

break;

}

this.currentMCCommand = mc.CommandID.ID;

IVsUIShell uiShell =

(IVsUIShell) ServiceProvider.GetService(typeof(SVsUIShell));

Guid clsid = Guid.Empty;

int result;

uiShell.ShowMessageBox(

0,

ref clsid,

"Test Tool Window Toolbar Package",

string.Format(CultureInfo.CurrentCulture, "You selected {0}", selection),

string.Empty,

0,

OLEMSGBUTTON.OLEMSGBUTTON\_OK,

OLEMSGDEFBUTTON.OLEMSGDEFBUTTON\_FIRST,

OLEMSGICON.OLEMSGICON\_INFO,

0,

out result);

}

}

测试菜单控制器

**1.** 生成解决方案然后开始调试。会打开实验实例。

**2.** 打开**视图 / 其他窗口**里的**Test ToolWindow**。

菜单控制器就会出现在工具窗口的工具栏里，显示了**MC Item 1**。

**3.** 点击菜单控制器箭头左边的按钮。

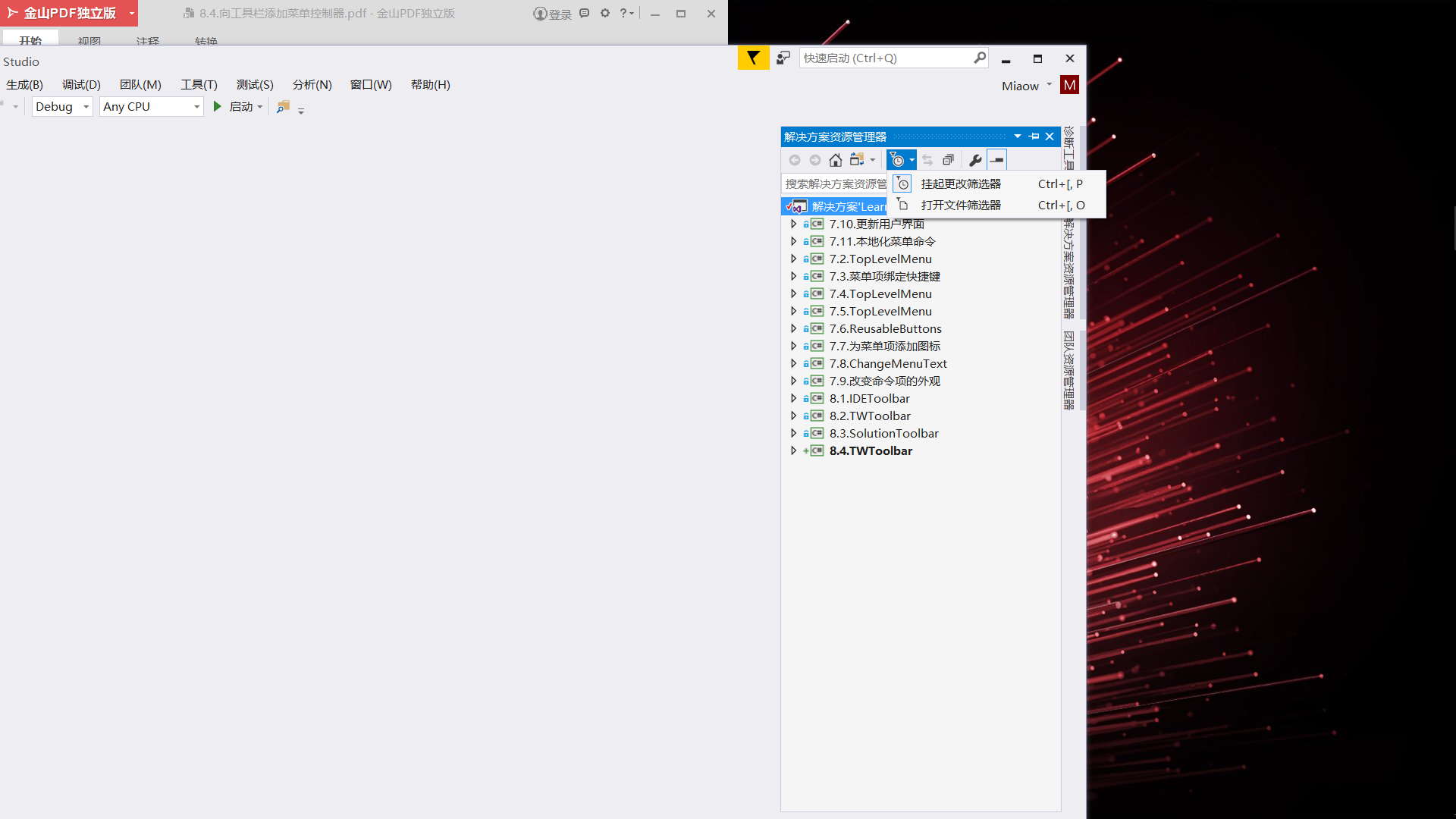
可以看到三项，第一项已被选中，图标上有会个高亮框。点击**MC Item 3**。

会出现一个显示**You selected Menu controller Item 3**的对话框。注意到显示的消息与菜单控制器里按钮的文本相符。菜单控制器现在会显示**MC Item 3**。

另见

[向工具窗口添加工具栏](https://docs.microsoft.com/zh-cn/visualstudio/extensibility/adding-a-toolbar-to-a-tool-window)

[添加一个工具栏](https://docs.microsoft.com/zh-cn/visualstudio/extensibility/adding-a-toolbar)

1. MIAOW注：本喵读书少，暂且翻译为菜单控制器（Menu Control），就是有下拉菜单箭头的按钮。如图：： [↑](#footnote-ref-0)
2. MIAOW注：在我的VS2017里是TWTestCommandPackage.cs，类名为TWTestCommandPackage，注意后面的代码里也要相应的改。 [↑](#footnote-ref-1)
3. MIAOW注：稍后会添加OnMCItemClicked方法和下一行的OnMCItemQueryStatus方法。 [↑](#footnote-ref-2)