

# 向解决方案资源管理器工具栏添加命令

原文

<https://docs.microsoft.com/zh-cn/visualstudio/extensibility/adding-a-command-to-the-solution-explorer-toolbar>

本文内容

准备

创建菜单命令项扩展

向解决方案资源管理器工具栏添加命令

控制按钮的可见性

另见

本指导演示如何在**解决方案资源管理器**工具栏里添加按钮。

Visual Studio中的工具栏或菜单上的任何命令都可以称为按钮。单击该按钮，将执行命令处理程序中的代码。通常，相关命令被组合在一起形成一个组。菜单或工具栏就是容纳这种组的容器。组里的命令的优先级决定其在菜单中或工具栏上显示的顺序。通过设置可见性，可以使按钮不在工具栏或菜单上显示。被列在.vsct文件的<VisibilityConstraints>里的命令只在处于有关语境时显示。但是不能对组设置可见性。

关于菜单、工具栏命令和.vsct文件，详见[命令、菜单和工具栏](#)。

注意：

使用 XML 命令表 (.vsct 文件) 来定义 VSPackage 里菜单和命令如何显示，不要使用命令表配置文件 (.ctc)。详见 [Visual Studio Command Table \(.Vsct\) Files](#)。

## 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](#)。

## 创建菜单命令项扩展

新建一个名为SolutionToolbar的VSIX项目。添加一个名为ToolbarButton的菜单命令项模板。关于如何做到这一点，详见[创建菜单命令项扩展](#)。

## 向解决方案资源管理器工具栏添加命令

这节演示如何在**解决方案资源管理器**工具栏里添加按钮。单击该按钮，将执行命令处理程序中的代码。

1. 在ToolbarButtonPackage.vsct文件里找到<Symbols>节点。<GuidSymbol>节点包含由模板生成的包含菜单组和命令。在这个节点里添加一个<IDSymbol>元素来声明一个新的组，用来包含要添加的命令。

```
<IDSymbol name="SolutionToolbarGroup" value="0x0190"/>
```

2. 在 <Groups>里的现有组项后面定义前一步声明的新组。

```
<Group guid="guidToolbarButtonPackageCmdSet" id="SolutionToolbarGroup" priority="0xF000">
  <Parent guid="guidSHLMainMenu" id="IDM_VS_TOOL_PROJWIN"/>
</Group>
```

把父对象的GUID:ID对设置为guidSHLMainMenu和IDM\_VS\_TOOL\_PROJWIN，这可以把组放到**解决方案资源管理器**的工具栏上。设置比较大的优先级值以放在其他组之后。

3. 改变 <Buttons>里自动生成的<Button>的父对象ID，使其指向上一步创建的组。修改后的<Button>元素看起来是像这样的：

```
<Button guid="guidToolbarButtonPackageCmdSet" id="ToolbarButtonId" priority="0x0100" typ=
"Button">
  <Parent guid="guidToolbarButtonPackageCmdSet" id="SolutionToolbarGroup" />
  <Icon guid="guidImages" id="bmpPicStrikethrough"1 />
  <Strings>
    <ButtonText>Invoke ToolbarButton</ButtonText>
  </Strings>
</Button>
```

4. 生成解决方案然后开始调试。会打开一个实验实例。

新的命令按钮会显示在**解决方案资源管理器**工具栏应该在现有按钮的右侧。按钮的图标是删除线。

5. 点击新的按钮。

会显示一个写着ToolbarButtonPackage Inside SolutionToolbar.ToolbarButton.MenuItemCallback()的对话框

## 控制按钮的可见性

本演练的这一部分展示了如何控制工具栏上按钮的可见性。在SolutionToolbar.vsct文件里设置<VisibilityConstraints>里的语境为一种或多种项目，这可以限制按钮只在设定的一种或多种项目打开时显示。

### 在一个或多个项目打开时显示按钮

1. 在ToolbarButtonPackage.vsct的<Buttons>节点里为<Button>元素添加两个命令标志( command flag )，就添加在 <Strings> 和 <Icons> 标签之间。

```
<CommandFlag>DefaultInvisible</CommandFlag>
<CommandFlag>DynamicVisibility</CommandFlag>
```

<sup>1</sup> MIAOW 注：这里也要改一下图标哦。

必须在设置DefaultInvisible和DynamicVisibility标志，这样<VisibilityConstraints>节点<sup>2</sup>里的项目才会生效。

2. 创建一个包含两个<VisibilityItem>项的<VisibilityConstraints>节点。就在关闭标签</Commands>之后创建如下新的节点。

```
<VisibilityConstraints>
  <VisibilityItem guid="guidToolBarButtonPackageCmdSet" id="ToolBarButtonId"
    context="UICONTEXT_SolutionHasSingleProject" />
  <VisibilityItem guid="guidToolBarButtonPackageCmdSet" id="ToolBarButtonId"
    context="UICONTEXT_SolutionHasMultipleProjects" />
</VisibilityConstraints>
```

每个VisibilityItem表示一种显示按钮的状况。若要应用多个条件，必须为同一按钮创建多个条目。

3. 生成解决方案然后开始调试。会打开一个实验实例。

此时**解决方案资源管理器**工具栏里还没有这个删除按钮。

4. 打开任何含有项目的解决方案。

删除按钮就会出现在工具栏现有按钮的右边。

5. 在**文件**菜单里点击**关闭解决方案**。工具栏上的按钮消失了。

加载VSPackage之前，按钮的可见性由Visual Studio控制。VSPackage载入后，VSPackage就控制了按钮可见性。详见[MenuCommands Vs. OleMenuCommands](#)。

## 另见

[命令、菜单和工具栏](#)

<sup>2</sup> MIAOW 注：<VisibilityConstraints>节点会在下一步创建。