

在菜单中添加子菜单

原文

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

本指导将通过在Visual Studio菜单栏添加菜单来说明如何显示如何添加子菜单到testmenu菜单。

子菜单就是指在其他菜单中的二级菜单，在其显示的名字右侧有个小箭头。点击子菜单可以展开其中的菜单项

我们将在Visual Studio菜单栏创建菜单并且在其中创建子菜单，并实现一些新的命令。

准备

启动Visual Studio 2015，不要从下载中心安装Visual Studio SDK。它在Visual Studio设置的可选功能里。你也可以稍后安装VS SDK。更多信息，参阅[Installing the Visual Studio SDK](#)。

为菜单添加子菜单

1. 按照[Adding a Menu to the Visual Studio Menu Bar](#)的步骤创建工程和菜单项。下面步骤中使用到的VSIX工程名为TopLevelMenu。
2. 打开TestCommandPackage.vsct文件。在<Symbols>里为子菜单的添加两个<IDSymbol>元素，其中一个表示子菜单的组，另一个表示子菜单里的命令项。这些都在名为guidTopLevelMenuCmdSet的<GuidSymbol>节点里，也就是包含表示top-level menu的<IDSymbol>元素的那个节点。

```
<IDSymbol name="SubMenu" value="0x1100"/>
<IDSymbol name="SubMenuGroup" value="0x1150"/>
<IDSymbol name="cmdidTestSubCommand" value="0x0105"/>
```

3. 在<Menus>里添加新创建的子菜单

```
<Menu guid="guidTestCommandPackageCmdSet" id="SubMenu" priority="0x0100" type="Menu">
  <Parent guid="guidTestCommandPackageCmdSet" id="MyMenuGroup"/>
  <Strings>
    <ButtonText>Sub Menu</ButtonText>
    <CommandName>Sub Menu</CommandName>
  </Strings>
</Menu>
```

<parent>里成对的GUID/ID说明在[Adding a Menu to the Visual Studio Menu Bar](#)里创建了这个菜单组 (menu group)，并且这个组是top-level menu里的。

4. 把步骤2里定义的menu group添加到<Groups>里，使它成为子菜单里的菜单组。

```
<Group guid="guidTestCommandPackageCmdSet" id="SubMenuGroup" priority="0x0000">
  <Parent guid="guidTestCommandPackageCmdSet" id="SubMenu"/>
</Group>
```

5. 在<Buttons>里添加一个<Button>元素用于步骤2在子菜单里创建的命令。

```
<Button guid="guidTestCommandPackageCmdSet" id="cmdidTestSubCommand" priority="0x0000" type="Button">
  <Parent guid="guidTestCommandPackageCmdSet" id="SubMenuGroup" />
  <Icon guid="guidImages" id="bmpPic2" />
  <Strings>
    <CommandName>cmdidTestSubCommand</CommandName>
    <ButtonText>Test Sub Command</ButtonText>
  </Strings>
</Button>
```

6. 生成解决方案并且调试，你将看到Visual Studio的实验实例。
7. 点击TestMenu可以看到名为Sub Menu的子菜单。点击使Sub Menu展开看到命令项Test Sub Command。注意此时点击Test Sub Command什么也不会发生。

添加一个命令

1. 打开TestCommand.cs并把以下命令ID添加到现有的命令ID之后。

```
public const int cmdidTestSubCmd = 0x105;
```

2. 添加子命令。找到命令的构造函数，把以下代码添加到AddCommand方法调用之后。

```
CommandID subCommandID = new CommandID(CommandSet, (int)TestCommandPackageGuids.cmdidTestSubCmd);
MenuCommand subItem = new MenuCommand(
    new EventHandler(SubItemCallback), subCommandID);
commandService.AddCommand(subItem);
```

SubItemCallback方法会在稍后传入。此时构造函数应该如下所示：

```
private TestCommand(Package package)
{
    if (package == null)
    {
        throw new ArgumentNullException("package");
    }
    this.package = package;
    OleMenuCommandService commandService = this.ServiceProvider.GetService(typeof(
    IMenuCommandService)) as OleMenuCommandService;
    if (commandService != null)
    {
        var menuCommandID = new CommandID(CommandSet, CommandId);
        var menuItem = new MenuCommand(this.MenuItemCallback, menuCommandID);
        commandService.AddCommand(menuItem);
        CommandID subCommandID = new CommandID(CommandSet, cmdidTestSubCmd);
        MenuCommand subItem = new MenuCommand(
            new EventHandler(SubItemCallback), subCommandID);
        commandService.AddCommand(subItem);
    }
}
```

3. 添加SubItemCallback()，当点击命令项时会调用这个方法。

```
private void SubItemCallback(object sender, EventArgs e)
{
    IVsUIShell uiShell = (IVsUIShell)this.ServiceProvider.GetService(
        typeof(SVsUIShell));
    Guid clsid = Guid.Empty;
    int result;
    uiShell.ShowMessageBox(
```

```
    0,  
    ref clsid,  
    "TestCommand",  
    string.Format(CultureInfo.CurrentCulture,  
        "Inside TestCommand.SubItemCallback()",  
        this.ToString()),  
    string.Empty,  
    0,  
    OLEMSGBUTTON.OLEMSGBUTTON_OK,  
    OLEMSGDEFBUTTON.OLEMSGDEFBUTTON_FIRST,  
    OLEMSGICON.OLEMSGICON_INFO,  
    0,  
    out result);  
}
```

4. 生成解决方案并调试，Visual Studio的实验实例会开始运行。
5. 在TestMenu菜单中点击Sub Menu然后再点击Test Sub Command。会显示一个消息对话框(message box)：“Test Command Inside TestCommand.SubItemCallback()”。