

改变命令项的外观

原文

<https://docs.microsoft.com/zh-cn/visualstudio/extensibility/changing-the-appearance-of-a-command>

本文内容

另见

改变菜单命令的外观

可以通过更改命令的外观来向用户提供反馈。比如，你可能希望命令在不可用时看起来不同。你可以让菜单里的命令看起来可用或不可用，显示或隐藏，选中或取消。

要改变命令项的外观，做其中的一步就可以了：

- 在命令表文件的命令定义里设定适当的标志。
- 使用 [OleMenuCommandService](#) 服务。
- 实现 [IOleCommandTarget](#) 接口并修改原始命令对象。

下面的步骤演示如何使用托管包框架（MPF）查找和更新命令的外观。

改变菜单命令的外观

1. 按照 [改变菜单项的文字](#) 的说明创建一个名为 `New Text` 的菜单项。

2. 打开 `ChangeMenuText.cs` 文件添加下面的 using 语句¹：

```
using System.Security.Permissions;
```

3. 打开 `ChangeMenuTextPackageGuids.cs` 文件添加以下语句：

```
public const string guidChangeMenuTextPackageCmdSet= "00000000-0000-0000-0000-00000000";2
// get the GUID from the .vsct file
```

4. 在 `ChangeMenuText.cs` 文件里用下面的代码替换 `ShowMessageBox` 方法：

```
private void ShowMessageBox(object sender, EventArgs e)
{
    var command = sender as OleMenuCommand;
    if (command.Text == "New Text")
        ChangeMyCommand3(command.CommandID.ID, false);
}
```

5. 从 [OleMenuCommandService](#) 对象获取更新的命令，然后为对象设置适当的属性。比如下面的方法使 `VSPackage` 命令集里的指定命令启用或不可用。下面的代码可以让名为 `New Text` 的菜单项点击后

¹ MIAOW 注：这句话似乎没用到啊。

² MIAOW 注：此处的 guid 从 .vsct 文件里面复制 `guidChangeMenuTextPackageCmdSet` 对应的值就可以了，别忘了复制过来把两边的大括号去掉。

³ MIAOW 注：这个方法稍后定义。

不可用。

```
public bool ChangeMyCommand(int cmdID, bool enableCmd)
{
    bool cmdUpdated = false;
    var mcs = this.ServiceProvider.GetService(typeof(IMenuCommandService))
        as OleMenuCommandService;
    var newCmdID = new CommandID(new Guid(ChangeMenuTextPackageGuids.
        guidChangeMenuTextPackageCmdSet), cmdID);
    MenuCommand mc = mcs.FindCommand(newCmdID);
    if (mc != null)
    {
        mc.Enabled = enableCmd;
        cmdUpdated = true;
    }
    return cmdUpdated;
}
```

6. 生成解决方案然后开始调试。这时就会出现Visual Studio的实验实例。
7. 在**工具**菜单里点击Invoke ChangeMenuText。这时它的名字还是Invoke ChangeMenuText，所以命令处理程序没有调用ChangeMyCommand()。
8. 现在在**工具**菜单里就可以看到New Text了。点击New Text。这个命令就变成了灰色的。

另见

[命令、菜单和工具栏](#)

[VSPackage是如何添加用户界面元素的](#)

[扩展菜单和命令](#)

[Visual Studio Command Table \(.Vsct\) Files](#)