

# 更新用户界面

原文<https://docs.microsoft.com/zh-cn/visualstudio/extensibility/updating-the-user-interface>

本文内容

另见

在实现命令之后，你可以添加代码来更新用户界面，刷新新命令的状态。

在典型的Win32应用程序中，可以不断查询命令集，当用户看到命令时调整更新单个命令的状态。但是Visual Studio Shell支持数量无限的VSPackage，大量的查询可能会降低性能，特别是托管代码和COM组件之间的交叉程序集查询。

## 更新 UI

1. 完成以下步骤之一：

- 调用[UpdateCommandUI](#)方法。

如下所示，可以从[SVsUIShell](#)服务里获取[IVsUIShell](#)接口。

```
void UpdateUI(Microsoft.VisualStudio.Shell.ServiceProvider sp)
{
    IVsUIShell vsShell = (IVsUIShell)sp.GetService(typeof(IVsUIShell));
    if (vsShell != null)
    {
        int hr = vsShell.UpdateCommandUI(0);
        Microsoft.VisualStudio.ErrorHandler.ThrowOnFailure(hr);
    }
}
```

如果[UpdateCommandUI](#)的参数不是0 ( `TRUE` )，则同步立即执行更新。我们建议你为这个参数传递0 ( `FALSE` )，来获得良好的性能。如果不需要缓存，在.vsct文件里创建命令时使用[DontCache](#)标志。不过请谨慎使用这个标志否则性能可能会下降。关于命令标志 ( command flags )，详见[命令标志 \( command flag \) 元素](#)。

- 在用就地激活模型 ( in-place activation model )<sup>1</sup>的含有ActiveX控件的VSPackage里，使用[UpdateUI](#)方法会很方便。[IVsUIShell](#)接口的[UpdateCommandUI](#)方法在功能上和[IOleInPlaceComponentUIManager](#)接口的[UpdateUI](#)方法是等同的。两者都使整个环境重新查询所有命令的状态。通常更新不会立即执行，会延迟到空闲时间再进行。为了保持良好的性能，Visual Studio Shell会缓存命令的状态。如果不需要缓存，在.vsct文件里创建命令时使用[DontCache](#)标志。不过请谨慎使用这个标志否则性能可能会下降。

注意，调用[IOleComponentUIManager](#)对象的[QueryInterface](#)方法可以获得一个[IOleInPlaceComponentUIManager](#)接口，也可以从[SOleComponentUIManager](#)服务里来获取这

<sup>1</sup> MIAOW 注：才疏学浅，没听说过这玩意，此处使用了百度翻译大法。有知道的大佬给我讲讲好吗，QQ3703781，感激不尽。

个接口。

## 另见

[VSPackage是如何添加用户界面元素的  
实现命令](#)