

创建菜单命令项扩展

本文内容

[准备](#)

[创建菜单命令](#)

[修改菜单命令处理程序](#)

[清理实验环境](#)

[部署你的扩展](#)

[后续步骤](#)

这个指导会说明如何创建用于启动记事本的菜单命令项扩展

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

1. 新建一个名为FirstMenuCommand的VSIX项目。VSIX项目在**新建项目**对话框下的Visual C# / Extensibility里。
2. 项目打开后添加一个名为FirstCommand的自定义命令项，如下所示。在**解决方案资源管理器**里右击项目节点，选择**添加/新建项**。在**添加新项**对话框选择Visual C# / Extensibility里的Custom Command。在窗口底部的名称栏里把文件名改为FirstCommand.cs。
3. 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。关于实验实例详见[实验实例](#)。

4. 在实验实例里打开**工具/扩展和更新**窗口。可以看到FirstMenuCommand。(如果在Visual Studio工作实例里打开的扩展和更新，则不会看到FirstMenuCommand)。

现在打开实验实例的**工具**菜单。应该看到Invoke FirstCommand命令项这里它只会显示一个消息框，“FirstCommandPackage Inside FirstMenuCommand.FirstCommand.MenuItemCallback()”。我们将在下一节中了解如何从这个命令实际启动记事本。

修改菜单命令处理程序

现在让我们更新命令处理程序来启动记事本。

1. 停止调试，回到Visual Studio的工作实例。打开FirstCommand.cs文件添加下面的using语句：

```
using System.Diagnostics;
```

2. 找到私有FirstCommand构造函数。这就是命令项连接到命令服务并指定为命令处理程序的地方。如下，改变命令处理程序的名称为StartNotepad：

```
private FirstCommand(Package package)
{
    if (package == null)
    {
        throw new ArgumentNullException(nameof(package));
    }

    this.package = package;

    OleMenuCommandService commandService = this.ServiceProvider.GetService(typeof(IMenuCommandService) as OleMenuCommandService);
    if (commandService != null)
    {
        CommandID menuCommandID = new CommandID(CommandSet, CommandId);
        // Change to StartNotepad handler.
        MenuCommand menuItem = new MenuCommand(this.StartNotepad, menuCommandID);
        commandService.AddCommand(menuItem);
    }
}
```

3. 删除MenuItemCallback方法并添加一个StartNotepad方法来启动记事本：

```
private void StartNotepad(object sender, EventArgs e)
{
    Process proc = new Process();
    proc.StartInfo.FileName = "notepad.exe";
    proc.Start();
}
```

4. 现在试试这段代码。当你开始调试项目并单击**工具/Invoke FirstCommand**，应该看到一个记事本被打开。

可以使用Process类的实例来运行任何可执行文件，不仅仅是记事本。比如试试calc.exe。

清理实验环境

如果你正在开发多个扩展，或者只是使用不同版本的扩展代码来调试，那么您的实验环境可能会不能正常工作。在这种情况下，应该运行重置脚本。它叫**Reset the Visual Studio 2015 Experimental Instance**，它是VisualStudio SDK的一部分。这个脚本删除实验环境里对你的扩展的所有引用，这样就可以从头开始。

可以通过以下两种方式之一使用这个脚本：

1. 在桌面或开始菜单（或开始屏幕）上找到**Reset the Visual Studio 2015 Experimental Instance**。
2. 在命令行运行以下命令：

```
<VSSDK installation>\VisualStudioIntegration\Tools\Bin\CreateExpInstance.exe /Reset /VSInstance=14.0 /RootSuffix=Exp && PAUSE
```

部署你的扩展

现在你的扩展已经可以按照你想要的方式运行了，是时候考虑如何分享给你的朋友和同事了。这很简单，前提是他们安装了Visual Studio 2015。你所要做的就把你生成的.vsix文件发给他们。（一定要在Release模式下生成解决方案。）

你可以在FirstMenuCommand 的bin目录找到这个扩展的.vsix文件。具体来说，如果生成的是Release版本，.vsix文件将在：

```
<code directory>\FirstMenuCommand\FirstMenuCommand\bin\Release\ FirstMenuCommand.vsix
```

你的朋友需要退出所有打开的Visual Studio，然后双击.vsix文件会启动VSIX Installer来安装这个扩展。安装时会复制相关文件到%LocalAppData%\Microsoft\VisualStudio\14.0\Extensions。

你的朋友再次启动Visual Studio时会在**工具/扩展和更新**里找到FirstMenuCommand扩展。他可以使用**扩展和更新**来卸载或禁用扩展。

后续步骤

以上仅展示了关于构建Visual Studio扩展的一小部分。下面简短（非常简短）列出关于Visual Studio扩展你还可以做的事：

1. 你可以用简单的菜单命令做更多的事情：
 - a. 添加你自己的图标：[为菜单项添加图标](#)
 - b. 改变菜单项的文字：[改变菜单项的文字](#)
 - c. 为菜单项添加快捷键：[菜单项绑定快捷键](#)
2. 添加不同种类的命令、菜单和工具栏：[扩展菜单和命令](#)
3. 添加工具窗口并扩展内置的Visual Studio工具窗口：[扩展和定制工具窗口](#)
4. 为现有的代码编辑器添加IntelliSense功能、代码建议和其他功能：[扩展编辑器和语言服务](#)
5. 向扩展中添加选项和属性页与用户设置：[扩展属性和属性窗口](#)和[扩展用户设置和选项](#)

其他类型的扩展需要更多的工作，比如创建一个新类型的项目（[扩展项目](#)），创建一个新的编辑器类型（[创建自定义的编辑器和设计器](#)），或者在独立模式（Isolated Shell）里实现你的扩展：[Visual Studio 独立模式（Isolated Shell）](#)。