

# 本地化菜单命令

原文<https://docs.microsoft.com/zh-cn/visualstudio/extensibility/localizing-menu-commands>

本文内容

对命令名进行本地化

对其他文本资源进行本地化

将本地化资源结合到项目中

另见

通过本地化VSPackage的.vsct文件和.resx文件，你可以为菜单和工具栏命令提供本地的文本。然后对项目文件进行升级来实现这些改变。

有关如何本地化安装体验的信息，见[对VSIX Packages进行本地化](#)。

## 对命令名进行本地化

1.在VSPackage里，菜单命令和工具栏按钮是在.vsct文件里定义的。

在**解决方案资源管理器**里把.vsct文件 *filename.vsct* 改为 *filename.en-US.vsct*。

2.用*filename.en-US.vsct*为每种要本地化的语言创建一个副本。

把这些副本重命名为 *filename.Locale.vsct*，其中*Locale*是特定文化的名字。有关文化名称的列表，见[支持的语言](#)。

这些 *filename.Locale.vsct*文件会包含本地化后的菜单文本。

3.打开每个 *filename.Locale.vsct*文件，对文本进行本地化。

- a. 针对特点语言修改[ButtonText](#)元素的值。
- b. 如果要使用本地化的图标，修改[Bitmap](#)的值让它指向目标文件。

下面的示例分别展示了英语和西班牙语的按钮文本，这是一个用来打开Family Tree Explorer工具窗口的按钮。

[FamilyTree.en-US.vsct]

```
<Button guid="guidLocalizedPackageCmdSet" id="cmdidFamilyTree" priority="0x0100" type="Button">
  <Parent guid="guidSHLMainMenu" id="IDG_VS_WNDO_OTRWNDWS1"/>
  <Icon guid="guidImages" id="bmpPic2" />
  <Strings>
    <CommandName>cmdidFamilyTree</CommandName>
    <ButtonText>Family Tree Explorer</ButtonText>
  </Strings>
</Button>
```

[FamilyTree.es-ES.vsct]

```
<Button guid="guidLocalizedPackageCmdSet" id="cmdidFamilyTree" priority="0x0100" type="Button">
  <Parent guid="guidSHLMainMenu" id="IDG_VS_WNDO_OTRWNDWS1"/>
  <Icon guid="guidImages" id="bmpPic2" />
  <Strings>
    <CommandName>cmdidFamilyTree</CommandName>
    <ButtonText>Explorar el arbol genealogico</ButtonText>
  </Strings>
</Button>
```

## 对其他文本资源进行本地化

资源 (.resx) 文件里定义了文本资源，而不是命令的名称。

1. 把VSPackage.resx重命名为VSPackage.en-US.resx。

2. 用VSPackage.en-US.resx为每种要本地化的语言创建一个副本。

把这些副本重命名为VSPackage.Locale.vsct，其中*Locale*是特定文化的名字。

3. 把Resources.resx重命名为Resources.en-US.resx。

4. 用Resources.en-US.resx为每种要本地化的语言创建一个副本。

把这些副本重命名为Resources.Locale.vsct，其中*Locale*是特定文化的名字。

5. 打开每个.resx文件按照字符串让它适合特定的语言和文化。下面的示例展示定义工具窗口标题栏的本地化资源。

[Resources.en-US.resx]

```
<data name="ToolWindowTitle" xml:space="preserve">
  <value>Family Tree Explorer</value>
</data>
```

[Resources.es-ES.resx]

```
<data name="ToolWindowTitle" xml:space="preserve">
  <value>Explorador del arbol genealogico</value>
</data>
```

## 将本地化资源结合到项目中

你须要修改assemblyInfo.cs文件和项目文件来结合本地化资源。

1. 在**解决方案资源管理器**中的**Properties**节点里，用编辑器打开assemblyinfo.cs或assemblyinfo.vb。

2. 添加以下条目。

```
[assembly: NeutralResourcesLanguage("en-US", UltimateResourceFallbackLocation.Satellite)]
```

这将美国英语作为默认语言。

3. 卸载项目。

4. 在编辑器里打开项目文件。

5.找到包含EmbeddedResource元素ItemGroup元素。

6.在调用VSPackage.en-US.resx的EmbeddedResource元素里，把ManifestResourceName元素替换为LogicalName元素，并把它设置成 VSPackage.en-US.Resources，如下所示。

```
<EmbeddedResource Include="VSPackage.en-US.resx">
  <MergeWithCTO>true</MergeWithCTO>
  <LogicalName>VSPackage.en-US.Resources</LogicalName>
</EmbeddedResource>
```

7.把刚才为VsPackage.en-US配置好的EmbeddedResource元素为每个需要本地化的语言复制一份，并把Include属性和LogicalName元素设到目标上，如下所示。

8.在每个本地化的VSCTCompile元素里添加一个指向Menus.ctmenu的ResourceName元素，如下所示。

```
<ItemGroup>
  <VSCTCompile Include="LocalizedPackage.es-ES.vsct">
    <ResourceName>Menus.ctmenu</ResourceName>
  </VSCTCompile>
</ItemGroup>
```

9.保存项目文件然后重新载入项目。

10.生成项目

这样创建了一个主程序集和对应每种语言的资源程序集。有关本地化部署过程的信息，见[对VSIX Packages进行本地化](#)。

## 另见

[扩展菜单和命令](#)

[MenuCommands Vs. OleMenuCommands](#)

[对应用程序进行全球化和本地化](#)