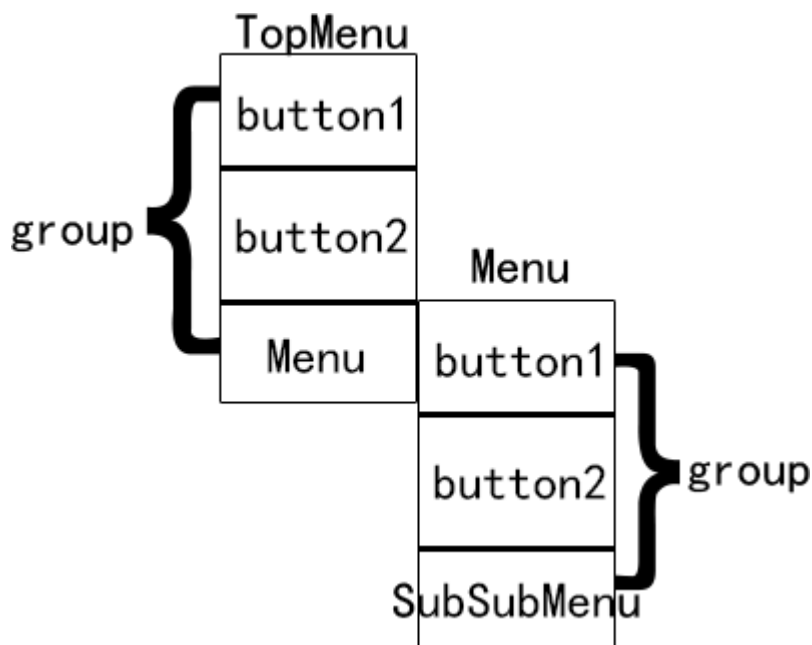


## VS Package

Visual Studio Command Table是一个文本文件，用来描述VSPackage包含的命令集。

要实现的功能：在顶层菜单添加一个菜单、三个子菜单



要实现这些功能，我们首先可以在 `.vsct` 文件中定义所需要的元素，比如Menu、Button、Group等。然后问题就是怎么把这些元素组织起来。

如上图：在顶级菜单(TopMenu)下可以有不带子菜单的button元素和带子菜单的Menu元素，我要想添加这些菜单到顶级菜单中，我可以把这些菜单先添加到一个组中，把这个组当成一个容器，然后把这些菜单和这个组关联起来（设置这些菜单的Parent为这个组）。然后设置这个组的Parent为顶级菜单（把这个组和顶级菜单关联起来）。这样这些菜单就添加到顶级菜单中了，要想把子菜单元素添加到菜单Menu中，可以先把这些子菜单都添加到另一个子菜单组中，然后把这个组和这个菜单Menu关联起来（设置子菜单组的Parent为Menu）。这样子菜单就添加到菜单中去了。这样就把这些元素组织起来了。

### 创建VSIX 项目

1.创建一个名为 `TopMenu` 的 VSIX 项目。在Visual C#/ Extensibility/VSIX Project。

2.当项目打开后时，添加一个名为自定义命令项模板TestCommand。在解决方案资源管理器，用鼠标右键单击项目节点并选择添加 / 新项。在添加新项对话框中，转到Visual C# / Extensibility，然后选择Custom Command。命令文件名称更改为TestCommand.cs。

在解决方案资源管理器，打开 TestCommandPackage.vsct文件。首先先在GuidSymbol下定义一个顶级菜单,定义它的name属性值为“TopMenu”,然后定义它的value属性值，如下：

```
<GuidSymbol name="guidTestCommandPackageCmdSet" value="{909f1bea-3103-4275-a894-be7ac0ac1f46}">
  <IDSymbol name="TopMenu" value="0x1021"/>
</GuidSymbol>
```

接着在标签下定义标签,它的id就为刚才设置的name属性值, 标签设置它的优先级, 然后设置它的Parent, 把它的Parent的guid设置为“guidSHLMainMenu”,id设置为“IDG\_VS\_MM\_TOOLSADDINS”, 使用guidSHLMainMenu和IDG\_VS\_MM\_TOOLSADDINS作为guid和id两个属性值的时候就不需要在Symbols元素里定义了, 因为着两个元素的值已经在stdidcmd.h和vsshld.h这两个文件里定义了, 而这两个文件在.vsct文件的开头就已经使用Extern元素引入了。所以就不需要我们去定义了。将这个标签放在标签之前, 这样这个顶级菜单就添加好了。

```
<Menus>
  <Menu guid="guidTestCommandPackageCmdSet" id="TopMenu" priority="0x700" type="Menu">
    <Parent guid="guidSHLMainMenu" id="IDG_VS_MM_TOOLSADDINS" />
    <Strings>
      <ButtonText>ReSharper</ButtonText>
      <CommandName>ReSharper</CommandName>
    </Strings>
  </Menu>
</Menus>
```

接下来我们就要添加菜单了, 在添加在菜单之前, 我们先在IDSymbol中定义一个组, 把这个组作为一个容器, 我们可以把其他的元素放到这个组中, 设置它的name属性值为“MyMenuGroup”,按照下面的添加到相应的标签中。

```
<GuidSymbol name="guidTestCommandPackageCmdSet" value="{909f1bea-3103-4275-a894-be7ac0ac1f46}">
  <IDSymbol name="MyMenuGroup" value="0x1020" />
  <IDSymbol name="TopMenu" value="0x1021"/>
</GuidSymbol>
```

接着我们就可以在Commands标签下定义这个Group,它的id就为刚才设置的name属性值, 组设置好之后, 怎么让它和顶级菜单关联起来呢, 在Group中可以添加Parent子元素 (Parent元素表示要将命令添加到哪个菜单下面), 把它的父节点设置为顶级菜单 (设置Group的Parent的id属性值为TopMenu),

```
<Group guid="guidTestCommandPackageCmdSet" id="MyMenuGroup" priority="0x0600">
  <Parent guid="guidTestCommandPackageCmdSet" id="TopMenu"/>
</Group>
```

这个容器设置好了, 我们就可以给里面添加其他元素了, 我们添加一个Button, 首先先在IDSymbol里定义一个Button。

```
<GuidSymbol name="guidTestCommandPackageCmdSet" value="{909f1bea-3103-4275-a894-be7ac0ac1f46}">
  <IDSymbol name="MyMenuGroup" value="0x1020" />
  <IDSymbol name="TopMenu" value="0x1021"/>
  <IDSymbol name="CommandId" value="0x0100" />
</GuidSymbol>
```

然后在Commands里面Buttons标签下修改这个Button,设置它的id为刚才定义的名字值, 并设置它的优先级 (这个优先级可以实现目标菜单的排列顺序, 优先级小的排列在上面)。接着让这个Button和Group关联起来 (设置Button的Parent为MyMenuGroup), 这样这些组件就一一关联起来了。

```
<Button guid="guidTestCommandPackageCmdSet" id="CommandId" priority="0x0100" type="Button">
  <Parent guid="guidTestCommandPackageCmdSet" id="MyMenuGroup" />
  <Icon guid="guidImages" id="bmpPic1" />
  <Strings>
    <CommandName>cmdidTestSubCommand</CommandName>
    <ButtonText>Why dotCover is disabled</ButtonText>
  </Strings>
</Button>
```

接下来实现带有子菜单的菜单，首先在IDSymbol定义一个菜单和一个子菜单组

```
<GuidSymbol name="guidTestCommandPackageCmdSet" value="{909f1bea-3103-4275-a894-be7ac0ac1f46}">
  <IDSymbol name="Menu" value="0x1100"/>
  <IDSymbol name="SubMenuGroup" value="0x1150"/>
  <IDSymbol name="cmdidTestSubCommand" value="0x0105"/>
</GuidSymbol>
```

然后把新创建的菜单添加到 `<Menus>` 部分。这里把这个新建的菜单加到MyMenuGroup这个组中。

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

添加子菜单组到 `<Groups>` 部分，然后设置它的Parent为Menu。子菜单到时候是加在这个子菜单组里面，然后把这个子菜单组加到刚才新建的菜单Menu下。

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

子菜单组创建好了之后，我们就可以添加一个新 `<Button>` 元素到 `<Buttons>` 部分，把这个 `Button` 元素的 `Parent` 设置为 `SubMenuGroup`，这样就把这个子菜单元素添加到这个子菜单组中了。这样子菜单和菜单就关联起来了

```
<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>
```

接下来怎么给每个命令添加功能呢。由于命令关联的操作是在点击命令的时候调用的，所有要实现该功能，要在Button元素加入子元素CommandName并设置它的值，然后在.cs文件中定义这个命令ID，在构造函数中把这个命令ID作为Command ID参数，然后为菜单添加命令处理程序(命令必须存在于命令表文件中)。把子菜单加到OleMenuCommandService服务（AddCommand）

```

public const int cmdidTestSubCmd = 0x105;

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);
    }
}

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);
}

```

