Adding a Menu to the Visual Studio Menu Bar

本文内容

Prerequisites

Creating a VSIX Project that has a Custom Command item template

Creating a Menu on the IDE Menu Bar

Building and Testing the Extension

See Also

This walkthrough shows how to add a menu to the menu bar of the Visual Studio integrated development environment (IDE). The IDE menu bar contains menu categories such as **File**, **Edit**, **View**, **Window**, and **Help**.

Before adding a new menu to the Visual Studio menu bar, consider whether your commands should be placed within an existing menu. For more information about command placement, see [Menus and Commands for Visual Studio](https://docs.microsoft.com/zh-cn/visualstudio/extensibility/ux-guidelines/menus-and-commands-for-visual-studio).

Menus are declared in the .vsct file of the project. For more information about menus and .vsct files, see [Commands, Menus, and Toolbars](https://docs.microsoft.com/zh-cn/visualstudio/extensibility/internals/commands-menus-and-toolbars).

By completing this walkthrough, you can create a menu named TestMenu that contains one command.

Prerequisites

Starting in Visual Studio 2015, you do not install the Visual Studio SDK from the download center. It is included as an optional feature in Visual Studio setup. You can also install the VS SDK later on. For more information, see [Installing the Visual Studio SDK](https://docs.microsoft.com/zh-cn/visualstudio/extensibility/installing-the-visual-studio-sdk).

Creating a VSIX Project that has a Custom Command item template

**1.** Create a VSIX project named TopLevelMenu. You can find the VSIX project template in the **New Project** dialog under Visual **C# / Extensibility**. For more information, see [Creating an Extension with a Menu Command](https://docs.microsoft.com/zh-cn/visualstudio/extensibility/creating-an-extension-with-a-menu-command).

**2.** When the project opens, add a custom command item template named **TestCommand**. In the **Solution Explorer**, right-click the project node and select **Add / New Item**. In the **Add New Item** dialog, go to Visual **C# / Extensibility** and select **Custom Command**. In the **Name** field at the bottom of the window, change the command file name to **TestCommand.cs**.

Creating a Menu on the IDE Menu Bar

To create a menu

**1.** In **Solution Explorer**, open TestCommandPackage.vsct.

At the end of the file, there is a <Symbols> node that contains several <GuidSymbol> nodes. In the node named guidTestCommandPackageCmdSet, add a new symbol, as follows:

<IDSymbol name="TopLevelMenu" value="0x1021"/>

**2.** Create an empty <Menus> node in the <Commands> node, just before <Groups>. In the <Menus> node, add a <Menu> node , as follows:

<Menus>

<Menu guid="guidTestCommandPackageCmdSet" id="TopLevelMenu" priority="0x700" type="Menu">

<Parent guid="guidSHLMainMenu"

id="IDG\_VS\_MM\_TOOLSADDINS" />

<Strings>

<ButtonText>TestMenu</ButtonText>

<CommandName>TestMenu</CommandName>

</Strings>

</Menu>

</Menus>

The guid and id values of the menu specify the command set and the specific menu in the command set.

The guid and id values of the parent position the menu on the section of the Visual Studio menu bar that contains the Tools and Add-ins menus.

The value of the CommandName string specifies that the text should appear in the menu item.

**3.** In the <Groups> section, find the <Group> and change the <Parent> element to point to the menu we just added:

<Groups>

<Group guid="guidTestCommandPackageCmdSet" id="MyMenuGroup" priority="0x0600">

<Parent guid="guidTestCommandPackageCmdSet" id="TopLevelMenu"/>

</Group>

</Groups>

This makes the group part of the new menu.

**4.** Find the Buttons section. Notice that the Visual Studio Package template has generated a Button element that has its parent set to MyMenuGroup. As a result, this command will appear on your menu.

Building and Testing the Extension

**1.** Build the project and start debugging. An instance of the experimental instance should appear.

**2.** The menu bar in the experimental instance should contain a **TestMenu** menu.

**3.** On the **TestMenu** menu, click **Invoke Test Command**.

A message box should appear and display the message "TestCommand Package Inside TopLevelMenu.TestCommand.MenuItemCallback()". This indicates that the new command works.

See Also

[Commands, Menus, and Toolbars](https://docs.microsoft.com/zh-cn/visualstudio/extensibility/internals/commands-menus-and-toolbars)