Adding a Command to the Solution Explorer Toolbar

本文内容

Prerequisites

Creating an Extension with a Menu Command

Adding a Button to the Solution Explorer Toolbar

Controlling the Visibility of a Button

See Also

This walkthrough shows how to add a button to the **Solution Explorer** toolbar.

Any command on a toolbar or menu is called a button in Visual Studio. When the button is clicked, the code in the command handler is executed. Typically, related commands are grouped together to form one group. Menus or toolbars act as containers for groups. Priority determines the order in which individual commands in a group appear in the menu or on the toolbar. You can prevent a button from being displayed on the toolbar or the menu by controlling its visibility. A command that is listed in a <VisibilityConstraints> section of the .vsct file appears only in the associated context. The visibility cannot be applied to groups.

For more information about menus, toolbar commands, and .vsct files, see [Commands, Menus, and Toolbar](https://docs.microsoft.com/zh-cn/visualstudio/extensibility/internals/commands-menus-and-toolbars)s.

Note

Use XML Command Table (.vsct) files instead of command table configuration (.ctc) files to define how menus and commands appear in your VSPackages. For more information, see [Visual Studio Command Table (.Vsct) Files](https://docs.microsoft.com/zh-cn/visualstudio/extensibility/internals/visual-studio-command-table-dot-vsct-files).

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 an Extension with a Menu Command

Create a VSIX project named SolutionToolbar. Add a menu command item template named **ToolbarButton**. For information about how to do this, see [Creating an Extension with a Menu Command](https://docs.microsoft.com/zh-cn/visualstudio/extensibility/creating-an-extension-with-a-menu-command).

Adding a Button to the Solution Explorer Toolbar

This section of the walkthrough shows how to add a button to the **Solution Explorer** toolbar. When the button is clicked, the code in the callback method is run.

**1.** In the ToolbarButtonPackage.vsct file, go to the <Symbols> section. The <GuidSymbol> node contains the menu group and command that was generated by the package template. Add an <IDSymbol> element to this node to declare the group that will hold your command.

XML

<IDSymbol name="SolutionToolbarGroup" value="0x0190"/>

**2.** In the <Groups> section, after the existing group entry, define the new group that you declared in the previous step.

<Group guid="guidToolbarButtonPackageCmdSet" id="SolutionToolbarGroup" priority="0xF000">

<Parent guid="guidSHLMainMenu" id="IDM\_VS\_TOOL\_PROJWIN"/>

</Group>

Setting the parent GUID:ID pair to guidSHLMainMenu and IDM\_VS\_TOOL\_PROJWIN puts this group on the **Solution Explorer** toolbar, and setting a high-priority value puts it after the other command groups.

**3.** In the <Buttons> section, change the parent ID of the generated <Button> entry to reflect the group that you defined in the previous step. The modified <Button> element should look like this:

<Button guid="guidToolbarButtonPackageCmdSet" id="ToolbarButtonId" priority="0x0100" typ="Button">

<Parent guid="guidToolbarButtonPackageCmdSet" id="SolutionToolbarGroup" />

<Icon guid="guidImages" id="bmpPicStrikethrough" />

<Strings>

<ButtonText>Invoke ToolbarButton</ButtonText>

</Strings>

</Button>

**4.** Build the project and start debugging. The experimental instance appears.

The **Solution Explorer** toolbar should display the new command button to the right of the existing buttons. The button icon is the strikethrough.

**5.** Click the new button.

A dialog box that has the message **ToolbarButtonPackage Inside SolutionToolbar.ToolbarButton.MenuItemCallback()** should be displayed.

Controlling the Visibility of a Button

This section of the walkthrough shows how to control the visibility of a button on a toolbar. By setting a context to one or more projects in the <VisibilityConstraints> section of the SolutionToolbar.vsct file, you restrict a button to appear only when a project or projects are open.

To display a button when one or more projects are open

**1.** In the <Buttons> section of ToolbarButtonPackage.vsct, add two command flags to the existing <Button> element, between the <Strings> and <Icons> tags.

<CommandFlag>DefaultInvisible</CommandFlag>

<CommandFlag>DynamicVisibility</CommandFlag>

The DefaultInvisible and DynamicVisibility flags must be set so that entries in the <VisibilityConstraints> section can take effect.

**2.** Create a <VisibilityConstraints> section that has two <VisibilityItem> entries. Put the new section just after the closing </Commands> tag.

<VisibilityConstraints>

<VisibilityItem guid="guidToolbarButtonPackageCmdSet"

id="ToolbarButtonId"

context="UICONTEXT\_SolutionHasSingleProject" />

<VisibilityItem guid="guidToolbarButtonPackageCmdSet"

id="ToolbarButtonId"

context="UICONTEXT\_SolutionHasMultipleProjects" />

</VisibilityConstraints>

Each visibility item represents a condition under which the specified button is displayed. To apply multiple conditions, you must create multiple entries for the same button.

**3.** Build the project and start debugging. The experimental instance appears.

The **Solution Explorer** toolbar does not contain the strikethrough button.

**4.** Open any solution that contains a project.

The strikethrough button appears on the toolbar to the right of the existing buttons.

**5.** On the **File** menu, click **Close Solution**. The button disappears from the toolbar.

The visibility of the button is controlled by Visual Studio until the VSPackage is loaded. After the VSPackage is loaded, the visibility of the button is controlled by the VSPackage. For more information, see [MenuCommands Vs. OleMenuCommands](https://docs.microsoft.com/zh-cn/visualstudio/extensibility/menucommands-vs-olemenucommands).

See Also

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