Changing the Appearance of a Command

原文

<https://docs.microsoft.com/zh-cn/visualstudio/extensibility/changing-the-appearance-of-a-command>

本文内容

See Also

To change the appearance of a menu command

You can provide feedback to your user by changing the appearance of a command. For example, you may want a command to look different when it is unavailable. You can make commands available or unavailable, hide or show them, or check or uncheck them on the menu.

To change the appearance of a command, perform one of these actions:

• Specify the appropriate flags in the command definition in the command table file.

• Use the [OleMenuCommandService](https://msdn.microsoft.com/en-us/library/microsoft.visualstudio.shell.olemenucommandservice(v=vs.140).aspx) service.

• Implement the [IOleCommandTarget](https://msdn.microsoft.com/en-us/library/microsoft.visualstudio.ole.interop.iolecommandtarget(v=vs.140).aspx) interface and modify the raw command objects.

The following steps show how to find and update the appearance of a command by using the Managed Package Framework (MPF).

To change the appearance of a menu command

**1.** Follow the instructions in [Changing the Text of a Menu Command](https://docs.microsoft.com/zh-cn/visualstudio/extensibility/changing-the-text-of-a-menu-command) to create a menu item named New Text.

**2.** In the ChangeMenuText.cs file, add the following using statement:

using System.Security.Permissions;

**3.** In the ChangeMenuTextPackageGuids.cs file, add the following line:

public const string guidChangeMenuTextPackageCmdSet= "00000000-0000-0000-0000-00000000";

// get the GUID from the .vsct file

**4.** In the ChangeMenuText.cs file, replace the code in the ShowMessageBox method with the following:

private void ShowMessageBox(object sender, EventArgs e)

{

var command = sender as OleMenuCommand;

if (command.Text == "New Text")

ChangeMyCommand(command.CommandID.ID, false);}

}

**5.** Obtain the command that you want to update from the [OleMenuCommandService](https://msdn.microsoft.com/en-us/library/microsoft.visualstudio.shell.olemenucommandservice(v=vs.140).aspx) object and then set the appropriate properties on the command object. For example, the following method makes the specified command from a VSPackage command set available or unavailable. The following code makes the menu item named New Text unavailable after it has been clicked.

public bool ChangeMyCommand(int cmdID, bool enableCmd)

{

bool cmdUpdated = false;

var mcs = this.ServiceProvider.GetService(typeof(IMenuCommandService))

as OleMenuCommandService;

var newCmdID = new CommandID(new Guid(ChangeMenuTextPackageGuids.guidChangeMenuTextPackageCmdSet), cmdID);

MenuCommand mc = mcs.FindCommand(newCmdID);

if (mc != null)

{

mc.Enabled = enableCmd;

cmdUpdated = true;

}

return cmdUpdated; }

}

**6.** Build the project and start debugging. The experimental instance of Visual Studio should appear.

**7.** On the **Tools** menu, click the **Invoke ChangeMenuText** command. At this point the command name is **Invoke ChangeMenuText**, so the command handler doesn't call ChangeMyCommand().

**8.** On the **Tools** menu you should now see **New Text**. Click **New Text**. The command should now be grayed out.

See Also

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

[How VSPackages Add User Interface Elements](https://docs.microsoft.com/zh-cn/visualstudio/extensibility/internals/how-vspackages-add-user-interface-elements)

[Extending Menus and Commands](https://docs.microsoft.com/zh-cn/visualstudio/extensibility/extending-menus-and-commands)

[Visual Studio Command Table (.Vsct) Files](https://docs.microsoft.com/zh-cn/visualstudio/extensibility/internals/visual-studio-command-table-dot-vsct-files)