

# How To Customize Context Menu in VS .NET

## How To Customize Context Menu in VS .NET

You already know that you can customize main menu and toolbars and you can place commands or macros on them. Unfortunately you cannot customize context (shortcut, right-click) menu this way in VS .NET 2002 and 2003. You simply cannot invoke context menu in Customize mode. You can do it easily in VS 2005.

For VS 2002/2003 we must use small trick. We create exact copy of context menu in VS main menu, right after the Help menu. This "mirror" menu is fully customizable. After we make changes to this copy, we copy them back to the context menu.

Add an item to Windows Explorer context (right-click) menu easily &ndash; How it may be done ?

Add items to Explorer Shell context menu easily with Explorer Context Menu. This powerful .Net component for your own, custom items appending to Windows Explorer Shell context menu will add all your custom application entries to the Explorer Shell context menu. This .Net component with full C# , C++ and VB.NET support include detailed C# / VB.NET samples, tutorials and support all you may need to add your entries to Explorer context menu :

- Add items to Windows Explorer Shell context menu to be shown on any Windows operating system (all operating systems are supported &ndash; XP, Vista, x64 , etc.)
- Add items to Windows Explorer Shell context menu to be shown in any way - with custom caption and your custom icon, as separator or sub-menu
- Add items of any types to Windows Explorer Shell context menu to be shown for all types of files or shown only for files of particular type (for example, only for .DOC , .MP3,.WMA,.AAC , .AVI files)
- Add your program entries to Windows Explorer Shell context menu, sub-menus, sub-menus of unlimited depth and add to Explorer context menu entries of all types

Windows Explorer Shell Context Menu - is a powerful .Net framework component that support all you may need to add all your items to Explorer context menu - in a fast and easy way. Add your entries to Windows Explorer Shell context menu right now &ndash; fast , easy and exactly as you prefer :

In more detail:

1. Create two simple macros contextMenuCreateEditable and contextMenuUpdateLastEdited. See their source code below.
2. Run contextMenuCreateEditable macro. This will create editable copy of context menu on main menu bar named \*Code Window\*.

Copy of context menu

3. Customize this menu in standard way. For example I want to add Comment and Uncomment selected lines commands to context menu as I use them a lot. Note that you may see more menu entries in this mirror menu than in original context menu. These are entries that are usually not shown in all contexts (for example breakpoint related).

Customizable "mirror" context menu

Also remember that add-ins may ignore your customizations. Because this method works only for Windows 95 / Windows 98 (not on XP, Vista, x64 - 64-bit Windows), to add items to Windows Explorer Shell context menu you should use, according to Microsoft guidelines, appropriate .Net component - Windows Explorer Shell Context Menu. For example our VBdocman .NET will always create its buttons (the last three in the picture above) after you restart VS even if you deleted them. Other addins may duplicate their buttons after your modifications. So I recommend you to temporarily unload all add-ins that insert buttons to the context menu during your customization (Tools - Add-in manager...)

4. You can update context menu with your changes after you are finished. Just run contextMenuUpdateLastEdited macro. It will also delete the temporary mirror menu.

That's all. It should work without any problems. You can still restore original context menu in the case you somehow destroy its contents. Simply run VS with /setup switch:

```
devenv.exe /setup
```

It will run in silent mode, no IDE will be shown.

Caution: Doing this will delete all customizations you have made to Visual Studio .NET menus, toolbars, and command groups. They are rebuilt according to the loaded add-ins and packages.

And here is the code:

PLAIN TEXT

```
Private editedContextMenu As Microsoft.Office.Core.CommandBar = Nothing
Private tmpContextMenu As Microsoft.Office.Core.CommandBar = Nothing
'''<summary>Creates editable mirror context menu
'''on main menu bar.</summary>
Sub contextMenuCreateEditable()
    Dim parent, menu, tmpMenu As Microsoft.Office.Core.CommandBar
    Dim tmpPopUp, contextPopUp As Microsoft.Office.Core.CommandBarPopup
    Dim ctrl, ctrl2 As Microsoft.Office.Core.CommandBarControl

    parent = DTE.CommandBars("MenuBar")
    menu = DTE.CommandBars("Code Window")
    tmpPopUp = DirectCast(parent.Controls.Add(Microsoft.Office.Core.MsoControlType.msoControlPopup, _
        ), Microsoft.Office.Core.CommandBarPopup)
    tmpPopUp.Caption = "*" & menu.Name & "*"
    tmpMenu = tmpPopUp.CommandBar
    For Each ctrl In menu.Controls
        ctrl2 = ctrl.Copy(tmpMenu)
        ctrl2.BeginGroup = ctrl.BeginGroup
    Next
    editedContextMenu = menu
    tmpContextMenu = tmpMenu
End Sub

'''<summary>Copies edited mirror menu to context menu.</summary>
Sub contextMenuUpdateLastEdited()
    Try
        Dim ctrl, ctrl2 As Microsoft.Office.Core.CommandBarControl

        If editedContextMenu Is Nothing Or tmpContextMenu Is Nothing Then
            Return
        End If

        For Each ctrl In editedContextMenu.Controls
            ctrl.Delete()
        Next
        For Each ctrl In tmpContextMenu.Controls
            ctrl2 = ctrl.Copy(editedContextMenu)
            ctrl2.BeginGroup = ctrl.BeginGroup
        Next
        CType(tmpContextMenu.Parent, Microsoft.Office.Core.CommandBarControl).Delete()
    Catch ex As System.Exception
    End Try
End Sub
```

If you don't know it yet, you can learn how to create and run macro.