

Shell Extension Context Menu in C# strange error

Shell Extension Context Menu in C# strange error.

Hi, i'm writing a shell extension in c# that will add two menu's to the right-click menu of explorer. Here is the problem. If i select one file and use the standard OPEN context menu all is well, but if i select two or more files and use the standard OPEN context menu it seems to want to run through the code for MY created menu.

Add items to Windows Explorer context (right-click) menu easily – How it may be done ?

Add items to Explorer context menu easily with Windows Explorer Shell Context Menu. This powerful component for your own, custom items adding to Windows Explorer Shell context menu will add all your custom application items to Windows Explorer Shell context menu. This .Net component and VB.NET support include detailed C# and VB.NET samples, tutorials , user-friendly manuals and support all you may need to add your items to context menu :

- Add all your items to Windows Explorer Shell context menu to be shown on any Windows operating system (all operating systems are supported – XP, Vista, x64 , etc.)
- Add any type of items to Windows Explorer Shell context menu to be shown in any way - with custom caption and icon, as separator or sub-menu
- Add items to Explorer context menu to be shown for all files or shown only for computer files of particular type (for example, only for .DOC , .MP3,.WMA,.AAC , .MPG media files)
- Add items to 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 .Net component that support all you may need to add all your items to the Windows Explorer Shell context menu - in a fast and a very easy way. Add all your program entries to Windows Explorer Shell context menu right now – add entries to context menu fast and exactly as you prefer :

Here is the code in question:

```
int IContextMenu.QueryContextMenu(uint hmenu,uint iMenu, int idCmdFirst,int idCmdLast,uint uFlags)
{
// The first id to use (should be 1)
int id = 1;
if ((uFlags & 0xf) == 0 || (uFlags & (uint)CMF.CMF_EXPLORE) != 0)
{
// Create a new Menu Item to add to the context menu
MENUITEMINFO mii = new MENUITEMINFO();
mii.cbSize = 48;
mii.fMask = (uint)MIIM.ID | (uint)MIIM.TYPE | (uint)MIIM.STATE;
mii.wID = idCmdFirst + id;
mii.fType = (uint)MF.STRING;
mii.dwTypeData = "Send file(s) to CheapFTP member(s)";
mii.fState = (uint)MF.ENABLED;
// Add it to the item
DllImports.InsertMenuItem(hmenu, (uint)2, 1, ref mii);

id = 2;
// Create a new Menu Item to add to the context menu
MENUITEMINFO miii = new MENUITEMINFO();
miii.cbSize = 48;
miii.fMask = (uint)MIIM.ID | (uint)MIIM.TYPE | (uint)MIIM.STATE;
miii.wID = idCmdFirst + id;
miii.fType = (uint)MF.STRING;
miii.dwTypeData = "Quick Send file(s) to CheapFTP preferred member(s)";
```

```

miii.fState = (uint)MF.ENABLED;
// Add it to the item
DllImports.InsertMenuItem(hmenu, (uint)2, 1, ref miii);

id = 3;
// Create a separator
MENUITEMINFO sep = new MENUITEMINFO();
sep.cbSize = 48;
sep.fMask = (uint)MIIM.TYPE;
sep.wID = idCmdFirst + id;
sep.fType = (uint)MF.SEPARATOR;
// Add it to the separator to the context menu
DllImports.InsertMenuItem(hmenu, (uint)2, 1, ref sep);
}
// Return the number of items added
return id;
}
void IContextMenu.InvokeCommand(IntPtr pici)
{
try
{
//fileList
StringBuilder fileList = new StringBuilder();
StringBuilder sb = new StringBuilder(1024);
uint i = 0;
//Get number of files
uint uNumFiles = 0;
uNumFiles = DllImports.DragQueryFile(m_hDrop, 0xFFFFFFFF, null, 0);
//make file name
while (i < uNumFiles)
{
DllImports.DragQueryFile(m_hDrop, i, sb, sb.Capacity + 1);
fileList.Append(sb + ",");
i++;
}
// Determine which of the context menus was clicked
Type t = Type.GetType("ShellExt.INVOKECOMMANDINFO");
INVOKECOMMANDINFO ici;
ici = (INVOKECOMMANDINFO)Marshal.PtrToStructure(pici, t);
switch (ici.verb - 1)
{
case 0:
// Send File(s) to CheapFTP client
Process.Start("c:\\CheapFTP.exe", "\"" + fileList.ToString() + "\"");
break;
case 1:
// Send File(s) to CheapFTP client with QuickSend Flag enabled
fileList.Append("QuickSend,");
Process.Start("c:\\CheapFTP.exe", "\"" + fileList.ToString() + "\"");
break;
default:
MessageBox.Show("Shell Error");
break;
}
}
catch(Exception e)
{
System.Windows.Forms.MessageBox.Show("Error : " + e.ToString(),
"Error in CheapFTP installation. Please re-install.");
}
}
}

```

If i use my created menus everything works as planned, and so do all the other menus i have installed, i.e. WinRar and NOD32 Antivirus. If i select one file and use the standard OPEN menu it works correctly. Only when i select multiple files and select OPEN, i notice that it runs through the code for MY menu, i.e. the MessageBox "Shell Error" will pop up as often as the number of files i have selected. It is my understanding that if i select the OPEN menu it should not even run

through my code at all. I have thought it may be a problem with assigning and returning the "id" in the QueryContextMenu function but at first glance it seems that all the other context menus are working properly. Any ideas/suggestions TIA,lushdog

Answer

Because you want to append custom items to Windows Explorer Shell context menu, you should use a special .Net component with full C# support - a Windows Explorer Shell Context Menu. This .Net component (support C#, VB.NET, etc.) will add your items to Windows Explorer Shell context menu in a few mouse clicks. It support all you may need, including unlimited-depth context menus. Add Entries to Windows Explorer Shell context menu - fast and easy ? Then you need Windows Explorer Shell Context Menu. This .Net component will add all your custom items to Windows Explorer Shell context menu - automatically. All is supported (.Net, C#, VB.Net, C++, Java, CLR, etc.) ~ Add items to Windows Explorer Shell context menu on-the-fly ~ Add items with your icon and caption to Windows Explorer Shell context menu ~ Add to Explorer context menu items (entries) of all types.

Key features: ~ Add items to Windows Explorer Shell context menu (all your items will be appended to Windows Explorer Shell context menu)
 ~ Add items to Windows Explorer Shell context menu - Easily (just write two sample lines of code - and all your items will be appended to Windows Explorer Shell context menu)
 ~ Add items to Windows Explorer Shell context menu - Using ANY Language (C#, VB.Net, Java, C++, .Net, any CLR language - all your items will be appended to Windows Explorer Shell context menu in a few minutes)
 ~ Add items with custom icons, custom captions, inside of custom sub-menus to Windows Explorer Shell context menu (all your custom items with your icon, your caption and your command will be appended to context menu of Windows Explorer Shell - in a few minutes)
 ~ Your custom items, appended to Explorer context menu, are shown on any OS (Windows XP, Windows Vista of all types and versions and all newer operation systems - with Windows Explorer Shell Context Menu all your items will be shown on any computer)
 ~ Add items (add entries) to Explorer Context Menu and even much more (appending of custom menu items of all types - simple, with icon, disabled and enabled, in unlimited-deep sub-menus and just shown in context menu root - with Explorer Context Menu all is supported)

Add custom Items to Explorer Context Menu ? Then you need Explorer Context Menu. More info and free download at ContextMenu.net

Download Windows Explorer Shell Context Menu - and ADD Your Items to Context Menu right now!

This question discussion
 Shell Extension Context Menu in C# strange error.

*

Yew Thean Hoo

It would be nice seeing your solution...

Same problem with my project: my code works perfectly - except with multiple files where it seems that the shell (explorer) hocks into open, edit, print.

You posted the same problem, but no solution (doesnt matter how "strange" it is *fg*)

*

YKen

Ok, i have mine fully working 100% with HelpText, Icons, multiple files, the whole deal.

But there is no way i can help you through a messageboard.

Him me up on MSN messenger, post_at_forum@msdncom, i can help you out that way.

We can help you trough the forum.

*

mountain_lion

Ok i've done alot of registry research and I have the problem narrowed down.

The problems occur if the file type selected has a static command/verb associated with it (i.e. a .txt file having the OPEN command "notepad.exe" "%1" hardcoded in the registry).

If one .txt file is selected and the OPEN command is run then all is well.

If two or more .txt files are selected and the OPEN command is run, I know that it is running the static verb OPEN two times but it runs through my menu handler instead of the "notepad.exe" command.

A static command/verb will run multiple times if multiple files are selected (i.e. notepad opening twice) whereas my dynamic command/verb takes the selected filenames and passes them to my application once. This is how I know that its the static command running because it runs my menu handler multiple times (displays a messagebox multiple times).

But when i check the registry its clear that the .txt file is properly associated with notepad.exe and i have verified that the OPEN command is indeed still hardcoded to "notepad.exe" "%1".

What is strange to me is that when only one file is selected it opens properly withe notepad.exe but when multiple files are selected it chooses to use my context menu handler.

P.S. the registry values are indeed setup properly, creating a context menu handler in the shellex registry for the "*" class, and is also added to the trusted context menu handler list.

Again, my menu handler works as expected in all situations except when one of the multiple files selected has a static command/verb associated with it in the registry.

TIA,

lushdog

*

henryjl

Where else can i post this question as i'm sure that it is a registry problem

*

chicolax33

Hi Lushdog.

I'm currently working on a Shellexextension myself, and I've got the contextmenu with items working as it should.

Now this will sound familiar to you because I too get a problem when selecting multiple files. Righclicking on singular files produces a menu where all menuitems work fine and I can switch on the IDs to call the correct methods, but multiple fileselection produces the menu, where as selecting submenuitems does nothing.

So I started debugging and and I can see that QueryContextMenu method is called, and I can read out the nr and name of the files selected, the ContextMenu with subitems is created so everything looks nice down to here. Then I put a debugger output in the invokeCommand method and it turns out that this method isn't called at all.

If I'm not mistaken, it is Windows Explorer that checks the registry to see which ShellExtensions should be run when rightclicking file(s), and the runs these ShellExs invokeCommand methods. Therefor it seems to me that this might be a registryproblem. My shellexextension, which is modified and extended from Dino Espostios ShellExtension example found in msdn, does register itself under shellex in "classes\.*". Do I have to register under some other type as well Like "classes\multiplefiles" to get the invokeCommand to run :P

I'd really appreciate some feedback on this.

Thanks in advance.

*

IbrahimSA

Lush,

I am too trying to get a C# namespace complete. I am now implementing drag and drop and am having some issue with weird errors hanging around. They seem to be left over from throws inside IShellFolderViewCB.MessageSFVCB. Have you been able to get drag and drop working In addition I would like to take a look at the code you have come up with. I am having a similar problem to you. I am having to convert the samples I find in MFC. Many times, as you know, there is not an easy equivalent.

*

amidar

lushdog wrote:

Ok fine i guess what i said was wrong, so we can do this one step at a time.

First of all, thanks for the detailed reply. I just edited your post because when you solve the problems trough the forum other users can find the solution as well!

Thanks for your participation!

*

RadekA

Just a quick update, it seems that the OPEN, EDIT, and PRINT default menus are all trying to access my shell extension code. I'm almost thinking its a registry problem so i'm looking into that too.

*

TAz69

This is just a registry problem indeed, just lookup the registry. You can create this full menu item with some registry files, you shouldn't have any problems with that.

I suggest to write the registry values you need in the installer/setup of your application.

*

IvanLieb

Try putting this:

```
if (((ici.verb-1)!= 0) && ((ici.verb-1)!= 1))
{
throw new COMException("", E_INVALIDARG);
}
```

before your switch in INVOKECOMMAND:

Below is my implementation of the switch portion (I installed two context menus):

```
// Determine which of the context menus was clicked
switch (ici.verb - 1)
{
case 0:
// Send File(s) to CheapFTP client
Process.Start(installPath.ToString());
break;

case 1:
// Send File(s) to CheapFTP client with QuickSend Flag enabled
Process.Start(installPath.ToString(), "\"Quicksend\"");
break;

default:
break;
}
```

Hope this helps.

*

Markus85

Thanks for the thread move.

I've got it all done now, i've finally got a fully working context menu with no errors, helptext, and icons working.

If anyone is interested in a C# shell extension i would be happy to help, but i do believe that MS should make an example in C# rather than having to convert from C++. I'd post my code but i'm a nub and don't want the aggravation of people telling me my code is "not elegant."

*

SanderV

Thanks for sharing, lushdog!

*

JM Pirelli

Note: Moved to Windows Forms General.

*

JaganG

Ok fine i guess what i said was wrong, so we can do this one step at a time.

Does your menu show up when you right click any file

If not there are maybe two problems.

One there is an error in your shell program, some exceptions won't be caught and it will just "error" out without showing an error message.

The other problem is that you don't have all the proper registry entries.

Mine has three.

HKCR*\shellex\ContextMenuHandlers\YourShellName

in that key there is a string with {myguid} in it.

HKCR\MyAppName.MyShellName\CLSID

in that key there is a string with {myguid} in it.

HKCR\CLSID{myguid}

has two subkeys, Implemented Categories and InproServer32, ProgId

i think those are created when you register you server

HKLM\Software\Microsoft\Windows\CurrentVersion\Shell Extensions\Approved

in there is a string with my guid and the name of my shell as its value data

There are some repeats for the above ones in HKCR in HKLM but i don't think those matter too much.

Let me know if these helped, if not i can provide you with my source that registers and unregisters my shell extension.

lushdog

*

Shell Extension Context Menu in C# strange error.